University of Zurich[UZH]

Zurich Open Repository and Archive
University of Zurich
Main Library
Strickhofstrasse 39
CH-8057 Zurich
www.zora.uzh.ch

Year: 2013

Calibration by correlation using metric embedding from non-metric similarities

Censi, Andrea ; Scaramuzza, Davide

Abstract: This paper presents a new intrinsic calibration method that allows us to calibrate a generic single-view point camera. From the video sequence obtained while the camera undergoes random motion, we compute the pairwise time correlation of the luminance signal for the pixels. We show that the pairwise correlation of any pixels pair is a function of the distance between the pixel directions on the visual sphere. This leads to formalizing calibration as a problem of metric embedding from non-metric measurements: we want to find the disposition of pixels on the visual sphere, from similarities that are an unknown function of the distances. This problem is a generalization of multidimensional scaling (MDS) that has so far resisted a comprehensive observability analysis and a generic solution. We show that the observability depends both on the local geometric properties as well as on the global topological properties of the target manifold. It follows that, in contrast to the Euclidean case, on the sphere we can recover the scale of the points distribution. We describe an algorithm that is robust across manifolds and can recover a metrically accurate solution when the metric information is observable. We demonstrate the performance of the algorithm for several cameras (pin-hole, fish-eye, omnidirectional).

DOI: https://doi.org/10.1109/TPAMI.2013.34

Posted at the Zurich Open Repository and Archive, University of Zurich
ZORA URL: https://doi.org/10.5167/uzh-71032
Journal Article
Accepted Version

Originally published at:
Censi, Andrea; Scaramuzza, Davide (2013). Calibration by correlation using metric embedding from non-metric similarities. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(10):2357-2370.
DOI: https://doi.org/10.1109/TPAMI.2013.34

Calibration by correlation using metric embedding from non-metric similarities

Andrea Censi, *Member, IEEE,* Davide Scaramuzza, *Member, IEEE*

Abstract—This paper presents a new intrinsic calibration method that allows us to calibrate a generic single-view point camera just by waving it around. From the video sequence obtained while the camera undergoes random motion, we compute the pairwise time correlation of the luminance signal for a subset of the pixels. We show that, if the camera undergoes a random uniform motion, then the pairwise correlation of any pixels pair is a function of the distance between the pixel directions on the visual sphere. This leads to formalizing calibration as a problem of *metric* embedding from *non-metric* measurements: we want to find the disposition of pixels on the visual sphere from similarities that are an *unknown* function of the distances. This problem is a generalization of multidimensional scaling (MDS) that has so far resisted a comprehensive observability analysis (can we reconstruct a metrically accurate embedding?) and a solid generic solution (how to do so?). We show that the observability depends both on the *local geometric properties* (curvature) as well as on the *global topological properties* (connectedness) of the target manifold. We show that, in contrast to the Euclidean case, on the sphere we can recover the scale of the points distribution, therefore obtaining a metrically accurate solution from non-metric measurements. We describe an algorithm that is robust across manifolds and can recover a metrically accurate solution when the metric information is observable. We demonstrate the performance of the algorithm for several cameras (pin-hole, fish-eye, omnidirectional), and we obtain results comparable to calibration using classical methods. Additional synthetic benchmarks show that the algorithm performs as theoretically predicted for all corner cases of the observability analysis.

Index Terms—intrinsic camera calibration, metric embedding, catadioptric cameras, pin-hole cameras, fish-eye cameras

1 Introduction

In many applications, from classic photogrammetry tasks to autonomous robotics, camera calibration is a necessary preliminary step before using the data [1]. Calibration is necessary even for off-the-shelf cameras, as the properties of an optical system may differ substantially from the manufacturer's specifications. *Extrinsic* camera calibration is concerned with recovering the pose (position and orientation) of the camera with respect to another camera, or another reference frame of interest. *Intrinsic* camera calibration is concerned with estimating the origin and direction of the line of sight of each pixel; this information is needed to put into correspondence the image of an object with the position of the object in the world. Some scientific applications require estimating other characteristics of the optical system, such as the point-spread function. Here, we focus on intrinsic camera calibration for central cameras.

In a central camera, the lines of sight of every pixel intersect in a single point. Therefore, the intrinsic calibration information consists of the direction of each pixel on the visual sphere ($\mathbb{S}^2$). If a camera is non central, then one needs to know, for each pixel, also its spatial position (in $\mathbb{R}^3$) in addition to its direction (in $\mathbb{S}^2$). A non-central camera can be approximated as a central camera only if the displacement of each pixel's origin is negligible with respect to the distance to the objects in the scene. This assumption is generally satisfied in applications such as robotics, but might not be satisfied for more uncommon applications and optical systems. A general description of how the properties of lenses, mirrors, and sensors contribute to the geometry of an optical system is outside of the scope of this paper; a recent tutorial is given by Sturm et al. [2].

Approaches and techniques for intrinsic calibration: Most widely used techniques for intrinsic camera calibration (from now on, simply "calibration") require that the camera's optics belongs to a restricted family. The model used for *pin-hole cameras* is parametrized by the center of projection, the focal length, and radial and tangential distortion, which account for the possibility of the image sensor being non perpendicular to the optical axis. Omnidirectional (*catadioptric*) cameras can be obtained by placing a mirror on top of a conventional camera, such that the optical axis coincides with the mirror's axis, or with fish-eye lenses (*dioptric*). The parameters of the model are the center of projection in image coordinates and the profile of the radial distortion. Several calibration software tools are available online as open source. The Matlab Calibration Toolbox [3] works for pin-hole cameras and implements a mix of techniques appeared in the literature [4], [5]. Other software [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16]. can be used for omnidirectional cameras. Puig et al. [17] give a practical and quantitative comparison.

In most of these methods, the user prints out a calibration pattern consisting of a black and white checker-

- *A. Censi is with the Control & Dynamical Systems department, California Institute of Technology. E-mail: andrea@cds.caltech.edu.*
- *D. Scaramuzza is with the AI Lab, Department of Informatics, University of Zurich. E-mail: davide.scaramuzza@ieee.org*

board, and collects several pictures of the pattern from different points of view. A semi-interactive procedure is used to identify the corners of the calibration pattern. Alternatively, the pattern might be autodetected, such as in OCamCalib [7], [18], [14]. Given the positions in image space of a known pattern, the software automatically solves for the calibration parameters. The algorithms rely on the fact that the pattern is known to lie on a plane, which allows recovering the parameters of the homography describing the world-to-image transformation, and that the nonlinear parts of the model, such as distortion, are simple enough that they can be recovered using generic nonlinear optimization. An alternative approach is recovering the calibration parameters as part of Euclidean reconstruction for moving cameras [19], [20].

Recent work has tried to improve flexibility and convenience by enlarging the family of optics considered, or making the calibration procedure more convenient. Grossberg and Nayar [21] describe a method for calibrating an arbitrary imaging system, in which the pixels are allowed to have an arbitrary configuration on the visual sphere, that is based on an active display. Hartley [22], Espuny and Gil [23], and Ramalingam et al. [24] describe techniques that do not require a known image pattern, but are based on known sensor motion.

Calibration in robotics: The last decade has seen the introduction of autonomous robotic systems where navigation and object recognition are performed with vision sensors. Precise calibration is needed to ensure the safety of operation. Moreover, calibration is considered for robots a "lifelong" activity that should be carried as autonomously as possible, as eventually any equipment degrades and must be re-calibrated over the course of its operating life; therefore, techniques that allow autonomous unsupervised calibration are especially valuable. Several calibration techniques have been designed to run autonomously embedded in a robotic architecture (e.g., [25], [26]); in these systems, a reduced set of camera calibration parameters (e.g., focal length) is treated as another state variable, and then estimated by a Bayesian filter together with the other states.

Calibration by correlation: In this paper, we will describe an approach to intrinsic camera calibration based exclusively on low-level statistics of the raw pixel streams, such as the inter-pixel correlation. To the best of our knowledge, Grossmann et al. [27] were the first to propose this idea for the problem of camera calibration, albeit they were inspired by work done in developmental robotics and related fields [28], [29], [30], [31].

The basic premise is that the statistics of the raw pixel stream contain information about the sensor geometry. Let $y_i(t)$ be the luminance perceived at the i-th pixel at time t. If we compare the sequences $\{y_i(t)\}_t$ and $\{y_j(t)\}_t$ for the i-th and j-th pixel, we expect to find that they are more similar the closer the two pixels are on the visual sphere. The geometry of the sensor can be recovered if one can find a statistics of the two sequences that is precisely a function of the pixels distances. More formally, let $s_i \in \mathbb{S}^2$ be the direction of the i-th pixel on the visual sphere, and let $d(s_i, s_j)$ be the geodesic distance on the sphere between the directions s_i and s_j. Let $S : \mathbb{R}^{\mathrm{T}} \times \mathbb{R}^{\mathrm{T}} \to \mathbb{R}$ indicate a real-valued statistics of two sequences of length T. For example, the statistics S can be the sample correlation, the mutual information, or any other information-theoretical divergence between two sequences, such as the "information distance" [27]. Define the *similarity* Y_{ij} between two pixels using S:

$$\mathrm{Y}_{ij} = S(\{y_i(t)\}_t, \{y_j(t)\}_t).$$

The assumption that must be verified for the method to work, which we will call the *monotonicity condition*, is that the similarity is a function f of the pixel distance:

$$\mathrm{Y}_{ij} = f(d(s_i, s_j)), \tag{1}$$

and that this f is monotonic, therefore, invertible.

Grossmann et al. assume to know the function f from a preliminary calibration phase, by using a camera with known calibration sensing the same scene as the camera being calibrated. If one knows f, the distances can be recovered from the similarities: $d(s_i, s_j) = f^{-1}(\mathrm{Y}_{ij})$. Two algorithms are described for recovering the pixel positions given the inter-pixel distances. The first algorithm is based on *multidimensional scaling* (which we will recall in the following sections) and solves for all pixel directions at the same time. The authors observe that this method is not robust enough for their data, and propose a nonlinear embedding method inspired by Sammon [32] and Lee et al. [33]. This method is iterative and places one pixel per iteration on the sphere, trying to respect all constraints with previously placed pixels.

Compared with the calibration methods mentioned above, the calibration-by-correlation approach is attractive because it does not require a parametric model of the camera geometry, control or knowledge of the instantaneous sensor motion, or the presence of interest points, patterns, or other structure in the scene.

However, the results reported by Grossmann et al. do not compare favorably with traditional methods. The authors focus their quantitative analysis mainly on the accuracy of the estimation and inversion of the function f. They find that, for the information distance, f is reliably invertible only for $d(s_i, s_j) \leq 30°$.[1] For large field of view, the estimated distributions appear significantly "shrunk" on the visual sphere.[2] Moreover, they find that, in practice, the function f is sensitive to the scene content; in their data, they find that applying the function f^{-1} estimated with a calibration rig to the data taken from a different camera leads to over-estimation of the angular distances.[3]

Paper outline: We start from the same premise of Grossmann et al., namely that it is possible to find a statistics of the pixel stream that depends on the pixel

1. Compare Fig. 6 in [27], which shows the graph of f as a function of distance; and Fig 8ab, which shows the error for estimating f^{-1}.

2. See Section 4.4.1 and Fig. 13 in [27]. Note the shrinkage of the distribution (no quantitative measure is given in the paper).

3. See Section 4.4.2 in [27].

distance. However, rather than assuming the function f known, we formulate a joint optimization problem, in which we solve for both the directions $\{s_i\}$ and the function f. In this way, there is no need for a preliminary calibration phase with a sensor of known geometry, but the problem becomes more challenging, requiring different analytic and computational tools.

Section 2 gives a formal description of the joint optimization problem. Section 3 discusses the conditions under which one can expect a monotonic relation between pixel distance and pixel statistics. We show that, if the camera undergoes uniform random motion, then necessarily all pairwise statistics between pixel values must depend on the pixel distance only. This suggests that a good way to collect data for camera calibration is to wave it around as randomly as possible, a theory we verify in practice.

Section 4 gives an observability analysis of the problem. The observability depends both on the manifold's *local geometric properties* (curvature) as well as on *global topological properties* (connectedness). In $\mathbb{R}^m$, the scale is not observable, but, surprisingly, it is observable in $\mathbb{S}^2$ and other spaces of nonzero curvature, which makes the problem more constrained than in Euclidean space.

Section 5 discusses the performance measures that are adequate for the problem. The Procrustes error (i.e., alignment up to rotations) is an intuitive choice, but it is not admissible because it is not invariant to all symmetries of the problem. We use the *Spearman score* as an admissible and observable performance measure.

Section 6 describes our algorithm, which is an extension of the classical Shepard-Kruskal (SK) algorithm [34], [35], [36], [37]. The major extension is an extra step necessary to recover the correct scale when it is observable; this step is critical for accurate calibration.

Section 7 discusses the experimental results for the case of camera calibration. The algorithm is evaluated for three different cameras: a pin-hole (45° FOV), a fish-eye (150° FOV), and an omnidirectional catadioptric camera (360° × 100° FOV). The results obtained are comparable with those obtained using conventional methods. Section 8 presents the results on a mix of real and synthetic datasets, which are meant to include all corner cases of the observability analysis. This shows that our algorithm is generic and works for other relevant cases in addition to the case of camera calibration.

Finally, Section 9 concludes the paper and discusses some possible directions for future work.

Supplemental materials: Appendix A contains the proofs that have been omitted for reasons of space. Appendix B contains the complete statistics and visualization for the benchmarks discussed. References to materials in Appendix A/B are written with the "A–" or "B–" prefix; e.g., Definition A-5. The hyperlinks link to an online copy of the documents. The attached multimedia materials include the source code to the algorithms and the intermediate processed results for the test cases discussed. Datasets and code are available at http://purl.org/censi/2012/camera_calibration.

2 Problem Formulation

Let $\mathcal{M}$ be a Riemannian manifold, and let d be its geodesic distance. We formalize the problem of metric embedding from non-metric measurements as follows.

Problem 1. Given a symmetric matrix $\mathbf{Y} \in \mathbb{R}^{n\times n}$, estimate the set of points $\mathcal{S} = \{s_i\}_{i=1}^n$ in a given manifold $\mathcal{M}$, such that $\mathrm{Y}_{ij} = f(d(s_i, s_j))$ for some (unknown) monotonic function $f : [0, \infty) \to \mathbb{R}$.

Without loss of generality, we assume the similarities to be normalized so that $-1 \leq \mathrm{Y}_{ij} \leq 1$ and $\mathrm{Y}_{ii} = 1$. This implies $f(0) = 1$, and that f is nonincreasing. For camera calibration, the manifold $\mathcal{M}$ will generally be the unit sphere $\mathbb{S}^2$; however, we formulate a slightly more generic problem. We will be especially interested in showing how the observability of the problem changes if $\mathcal{M}$ is chosen to be $\mathbb{S}^1$ (the unit circle) or $\mathbb{R}^m$ instead of $\mathbb{S}^2$.

If the function f was known, it would be equivalent to know directly the matrix of distances. The problem of finding the positions of a set of points given their distance matrix is often called "metric embedding". In the Euclidean case ($\mathcal{M} = \mathbb{R}^m$), the problem is classically called *Multidimensional Scaling* (MDS), and was first studied in psychometry in the 1950s. Cox and Cox [38] describe the statistical origins of the problem and give an elementary treatment, while France and Carroll [37] give an overview of the algorithmic solutions.

The scenario described in Problem 1 is sometimes called *non-metric* multidimensional scaling. The word "non-metric" is used because the metric information, contained in the distances $d(s_i, s_j)$, is lost by the application of the unknown function f. In certain applications, it is not important for the reconstructed points to be recovered accurately. For example, in psychometry, one might use these techniques essentially for visualization of high-dimensionality datasets; in that case, one only wants a *topologically correct* solution. If that is the case, one can just choose an arbitrary $\tilde{f}$ different from the true f; as long as $f(0) = \tilde{f}(0)$, the results will be topologically correct. However, in the camera calibration setting, we are explicitly interested in obtaining a *metrically accurate* solution. Note that Problem 1 is a chicken-and-egg problem in the two unknowns f and $\{s_i\}_{i=1}^n$: knowing the function f, one can estimate the distances as $f^{-1}(\mathrm{Y}_{ij})$, and use standard MDS to solve for $\{s_i\}_{i=1}^n$; conversely, knowing the distances, it is trivial to estimate f. But is it possible to estimate both at the same time? To the best of our knowledge, there has not been any claim about whether *accurate metric embedding from non-metric measurements* is possible. In this paper, we will show that the answer depends on the properties of the manifold $\mathcal{M}$. Specifically, while for $\mathbb{R}^m$ the scale is not observable, we show that accurate metric embedding is possible for $\mathbb{S}^2$. Consequently, it is possible to calibrate a camera from any statistics that respects the monotonicity condition (1), even if the function f is a priori unknown.

Other problems with an equivalent formalization: We are motivated to solve the problem in a slightly more generic way than what strictly needed for camera calibration because some problems in different fields have an equivalent formalization. In developmental robotics [29], [30], [31], a common scenario is that an agent starts from zero knowledge of its sensors, and its first concern is to recover the geometry of the sensor (possibly a camera, but also a range-finder or other robotic sensor) by considering simple statistics of the sensor streams. In *sensor networks* (see, e.g., [39]), one basic problem is localizing the nodes in space based on relative measurements of wi-fi strength. Assuming the signal is a function of the distance, one arrives to the same formalization, using $\mathbb{R}^n$ as the target manifold. More generally, this formalization covers many embedding problems in machine learning, where the data is assumed to be in a metric space, but the available similarities, perhaps obtained by comparing vectors of *features* of the data, cannot be interpreted directly as distances in the original metric space.

3 When is similarity a function of the pixels distance?

The basic assumption of this paper is that it is possible to find a statistics of the pixel luminance that satisfies the monotonicity condition (1). We state a result that guarantees that *any* pairwise statistics is (asymptotically) a function of the distance between the pixels, if the camera undergoes uniformly random motion, in the sense that the camera's orientation (a rotation matrix $\mathbf{R}$) is uniformly distributed in $\mathsf{SO}(3)$.

Proposition 2. *If the probability distribution of the camera orientation $\mathbf{R}$ is uniform in $\mathsf{SO}(3)$, the expectation of a function of the luminance of two pixels depends only on the pixel distance: for all functions $g : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, there exists a function $f : \mathbb{R}^+ \to \mathbb{R}$, such that*

$$\mathbb{E}\{g(y(s_i), y(s_j))\} = f(d(s_i, s_j)),$$

where $\mathbb{E}\{\cdot\}$ denotes the expectation with respect to $\mathbf{R}$.

Proof: See Section A-2 of Appendix A. □

In particular, this is valid for the correlation between pixel values, as the correlation can be written as $\mathsf{corr}(y_i, y_j) = \mathbb{E}\{g(y(s_i), y(s_j))\}$ with $g(y(s_i), y(s_i)) = (y(s_i) - \overline{y})(y(s_i) - \overline{y})$. Most other similarity statistics can be written in the same fashion.

When is similarity monotonic?

Proposition 2 ensures that (1) holds for some function f, but it does not ensure that such function f is monotone. To find conditions that guarantee that f is monotone it is necessary to introduce some model of the environment. Essentially, f might not be monotone if there is some long-range "structure" in the environment. In the case of the correlation, the statistics of natural images allow concluding that f is monotone for small pixel distances [40]. We describe an artificial counter example in which f is not monotone for larger distances.

Example 3. An omnidirectional camera is suspended in the middle of a room shaped like a parallelepiped with the base of size $L \times L$ and height $H \gg L$ (Fig. 1). The projections of ceiling and floor on the visual sphere are contained in a spherical cap of radius $\delta = 2 \arccos\left(H/\sqrt{H^2 + L^2}\right)$. For example, $L = 5$m and $H = 10$m give $\delta \simeq 28°$. This implies that two pixels observing the ceiling at the same time cannot be more than $28°$ apart. Assume that the floor and the ceiling are painted of a uniform white, and the walls have very intricate black-white patterns, well approximated by white noise. We let the camera undergo random rotational motion, and we compute the correlation of the pixel luminance. Consider two pixels at distance $d(s_i, s_j) = 60°$. They cannot look both at the ceiling at the same time, because the apparent size of the ceiling is $\delta = 28°$. There are three possibilities: 1) they are both looking at the walls; 2) one is looking at the walls, another at the ceiling; 3) one is looking at the walls, another at the floor. In all cases, one is looking at the white noise on the walls. Therefore, the correlation at distance $60°$ is 0: $f(60°) = 0$. Consider two pixels at distance $180°$. There are two possibilities: 1) they are both looking at the walls; 2) one looks at the ceiling, the other at the floor. Because floor and ceiling have the same color, the luminance of these two pixels is positively correlated: $f(180°) > 0$. Therefore, the function f is not monotonic, because $f(0°) = 1$, $f(60°) = 0$, and $f(180°) > 0$.

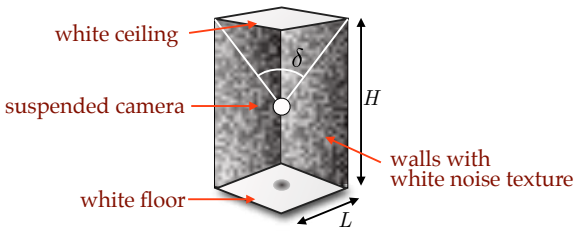


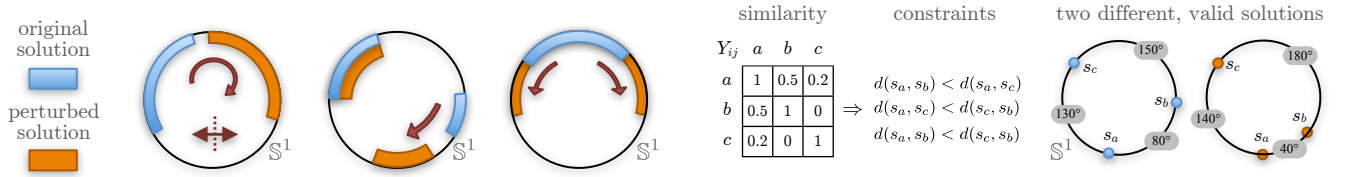
Fig. 1: Environment used in Example 3.

4 Observability analysis

A *symmetry* of an estimation problem is any joint transformation of the unknowns (in this case, the directions $\mathcal{S} = \{s_i\}$ and the function f) that does not change the observations (in this case, the similarities Y_{ij}). Studying the observability of the problem means describing what symmetries are present. In this section, we first give a tour of the symmetries of this problem, before presenting the main result in Proposition 8.

Isometries: It is easy to see that the similarities Y_{ij} are preserved by the isometries of the domain $\mathcal{M}$.

Definition 4. An *isometry* of $\mathcal{M}$ is a map $\varphi : \mathcal{M} \to \mathcal{M}$ that preserves distances: $d(\varphi(s_i), \varphi(s_j)) = d(s_i, s_j)$. We denote the set of all isometries of $\mathcal{M}$ by $\mathsf{Isom}(\mathcal{M})$.



(a) Isometries (b) Sliding (c) Warping (d) Wiggling: an extreme case, with only three points.

Fig. 2: Symmetries of the estimation problem, illustrated in the case $\mathcal{M} = \mathbb{S}^1$. (a) Isometries (for $\mathbb{S}^1$, rotations and reflections) are unobservable because they preserve distances between points. (b) If f is non invertible on the whole domain, disconnected components of $\mathcal{S}$ can move isometrically independently from each other; we call this "sliding". (c) In manifolds with zero curvature (e.g., $\mathbb{R}^n$ and $\mathbb{S}^1$, but not $\mathbb{S}^2$) the scale is not observable; formally, a linear warping (Definition 6) does not violate the problem constraints. (d) If the set of points is finite, the constraints are not violated by small perturbations of the points, called "wigglings".

Sliding: This symmetry exists if f is not informative enough. Define the "informative radius" of f as follows.

Definition 5. For a function $f : \mathbb{R}^+_\circ \to \mathbb{R}$, let $\mathsf{infr}(f)$ be the maximum r such that f is invertible in $[0, r]$.

If $\mathcal{S}$ has two components distant more than $\mathsf{infr}(f)$ from each other, then one can be isometrically moved independently of the other without changing the observations (Fig. 2b); we call this "sliding".

Linear warping: We define a *linear warping* as a map that scales the inter-point distances uniformly by a constant.

Definition 6. A linear warping of $\mathcal{M}$ is a map $\varphi_\alpha : \mathcal{M} \to \mathcal{M}$ such that, for some $\alpha > 0$, for all $s_1, s_2 \in \mathcal{M}$,

$$d(\varphi_\alpha(s_1), \varphi_\alpha(s_2)) = \alpha\, d(s_1, s_2),$$

If a linear warping exists, then it is a symmetry of the problem. In fact, suppose that $(f, \{s_i\})$ is a solution of the problem. Construct another solution $(f', \{s'_i\})$, with $f' = \frac{1}{\alpha} f$ and $s'_i = \varphi_\alpha(s_i)$. We would not be able to distinguish between these two solutions, as they would give the same observations.

Wiggling: A peculiar aspect of Problem 1 is that the unknowns $\mathcal{S} = \{s_i\}_{i=1}^n$ live in a continuous space $\mathcal{M}$, but the observations $\mathrm{Y}_{ij} = f(d(s_i, s_j))$ are actually equivalent to a set of *discrete inequalities*, a fact which is very well explained by Agarwal et al. [41]. In fact, because the function f is completely unknown (the only constraint being its monotonicity), all that we can infer about the inter-point distances from the matrix $\mathbf{Y}$ is their ordering: for all $(i, j), (k, l)$, if $\mathrm{Y}_{ij} \gtreqless \mathrm{Y}_{kl}$, then we can infer $d(s_i, s_j) \gtreqless d(s_k, s_l)$, but nothing more. Therefore, the sufficient statistics in the matrix Y_{ij} is the *ordering* of the entries, not their specific *values*. This raises the question of whether precise metric reconstruction is possible, if the available observations are a set of discrete inequalities. In fact, given a point distribution $\{s_i\}$ of n points, the position of the generic point s_j is constrained by n^2 inequalities (many of which redundant). Inequalities cannot constrain a specific position for s_j in the manifold $\mathcal{M}$; rather, they specify a small finite area in which all constraints are satisfied. Therefore, for each solution, an individual point has a small neighborhood in which it is free to "wiggle" without violating any constraint. In general, we call these perturbations "wiggling"[4]:

Definition 7. A *wiggling* of a set $\{s_i\} \subset \mathcal{M}$ is a map $\varphi : \mathcal{M} \to \mathcal{M}$ that preserves the distance ordering:

$$d(s_i, s_j) < d(s_k, s_l) \Leftrightarrow d(\varphi(s_i), \varphi(s_j)) < d(\varphi(s_k), \varphi(s_l)).$$

The size of the allowed wiggling decreases with the density of points; for n points uniformly distributed in $\mathcal{M}$, one can show that the average wiggling space is in the order of $o(1/n)$ per single point (i.e., keeping the others fixed). In the limit as the points become dense, it is possible to show that wiggling degenerates to rigid linear warpings. For very sparse distributions, the effect of wiggling can be quite dramatic (Fig. 2d).

Main result: The following proposition establishes which of the previously described symmetries are present in the problem, as a function of the geometric properties of the space $\mathcal{M}$, the point distribution, and the function f.

Proposition 8. *Assume the set $\mathcal{S} = \{s_i\}$ is an open subset of $\mathcal{M}$ whose closure has only one connected component. Let the available measurements be $\mathrm{Y}_{ij} = f(d(s_i, s_j))$, where $f : \mathbb{R}^+_\circ \to \mathbb{R}$ is a monotone function with $\mathsf{infr}(f) > 0$. Then:*

- *If $\mathcal{M}$ has nonzero curvature (e.g., $\mathbb{S}^2$), then it is possible to recover f exactly, and $\mathcal{S}$ up to isometries.*
- *If $\mathcal{M}$ has zero curvature:*
  - *If $\mathcal{M}$ is simply connected (e.g., $\mathbb{R}^m$), then it is possible to recover f only up to scale, and $\mathcal{S}$ up to isometries plus a "linear warping" (Definition 6).*
  - *If $\mathcal{M}$ is not simply connected (e.g., $\mathbb{S}^1$), the scale can be recovered if $\mathsf{infr}(f)$ is large enough.*
    For $\mathbb{S}^1$, this happens if

$$\mathsf{rad}(\mathcal{S}) + \mathsf{infr}(f) \geq \pi, \tag{2}$$

    where $\mathsf{rad}(\mathcal{S})$ is the radius of $\mathcal{S}$ (Definition A-1).

The observability breaks down as follows:

- *"Sliding" occurs if $\mathcal{S}$ has multiple components with Hausdorff distance greater than $\mathsf{infr}(f)$.*
- *If $\mathcal{S}$ has a finite number of points, there is a "wiggling" uncertainty, in the order of $o(1/n)$ for uniform distributions.*

4. By these definitions, isometries $\subseteq$ linear warping $\subseteq$ wiggling; studying when the inclusions of these transformations classes is the basis of the observability analysis reported in Appendix A.

TABLE 1: Observability classes

class	*space*	*curvature*	*extra assumptions*	*symmetries*
A	$\mathbb{S}^{\geq 2}$	> 0	-	wigglings, isometries
B	$\mathbb{S}^1$	0	$\mathsf{rad}(\mathcal{S}) + \mathsf{infr}(f) \geq 2\pi$	wigglings, isometries
C	$\mathbb{S}^1$	0	$\mathsf{rad}(\mathcal{S}) + \mathsf{infr}(f) < 2\pi$	wigglings, isometries, linear warpings
D	$\mathbb{R}^n$	0	-	wigglings, isometries, linear warpings
A	$\mathbb{H}^n$	< 0	-	wigglings, isometries

Proof: See Section A-3 in Appendix A. □

The results are summarized in Table 1 and the various observability classes are labeled A–D for later reference. We note that the observability results depend both on the *local geometrical properties* of the space (curvature) as well as the *global topological properties* (connectedness).

5 Measuring performance

The performance of an algorithm must be measured in a way compatible with the observability of the problem. We expect an error score to be *invariant*, meaning that it is conserved by the symmetries of the problem. If a score is not invariant, it is measuring something that is not possible to estimate. We expect an error score to be *complete*, meaning that it is minimized only by the solutions of the problem. If an error score is not complete, it cannot be used to distinguish solutions from non solutions. Finally, we wish the error score to be *observable*, in that it can be computed from the data, without the ground truth.

Distances-based performance measures: In our case, several classical error measures, widely used in other contexts, do not satisfy all these properties. The Procrustes error is defined as the mean distance between the solution $\{s_i\}_{i=1}^n$ and the ground truth $\{\overline{s}_i\}_{i=1}^n$, after choosing the best isometry that makes the two sets overlap [42].

Definition 9. The *Procrustes error* e_{pr} is defined as:

$$e_{\text{pr}}(\{\overline{s}_i\}, \{s_i\}) \triangleq \min_{\varphi \in \mathsf{Isom}(\mathcal{M})} \frac{1}{n} \sum_{i=1}^{n} d(\overline{s}_i, \varphi(s_i)). \tag{3}$$

This error score is unsuitable in our case, because, while it is invariant to isometries, it is not invariant to the other symmetries, namely linear warpings and wigglings. This means that, if we are considering an instance of the problem where the scale is not observable, using the Procrustes error can produce misleading results (we will show this explicitly in Section 8). Moreover, there is the problem that not all points contribute equally to this performance measure. When aligning the two points sets, the points near the center of the distribution will be always more aligned, and the errors will accumulate for the points at the borders of the distribution. To eliminate this problem, we can consider the error on the inter-point distances rather than the absolute position of the points.

Definition 10. The *mean relative error* e_{r} is the mean error between the inter-point distances:

$$e_{\mathsf{r}}(\{\overline{s}_i\}, \{s_i\}) \triangleq \frac{1}{n^2} \sum_{i,j=1}^{n} |d(\overline{s}_i, \overline{s}_j) - d(s_i, s_j)|. \tag{4}$$

This error function is still invariant to isometries, does not need an optimization problem to be solved, and all pairs of points contribute equally. Moreover, it can be easily modified to be invariant to linear warpings: because linear warpings scale the distances uniformly, we achieve invariance by optimizing over an unknown scale.

Definition 11. The *mean scaled relative error* e_{sr} is the relative error after the optimal warping:

$$e_{\mathsf{sr}}(\{\overline{s}_i\}, \{s_i\}) \triangleq \min_{\alpha > 0} \frac{1}{n^2} \sum_{i,j=1}^{n} |d(\overline{s}_i, \overline{s}_j) - \alpha\, d(s_i, s_j)|. \tag{5}$$

This is invariant to warpings, but still not invariant to wigglings.

Spearman-correlation-based performance measures: We introduce the *Spearman score*: an invariant, complete, and observable score for all observability classes. It is based on the idea of *Spearman correlation*, which measures a possibly *nonlinear* dependence between two variables, in contrast with the usual correlation, which can only assess linear dependence. The Spearman correlation is a common tool in applied statistics, but it is not widely used in engineering. The idea is that, to assess nonlinear relations, we should consider not the value of each datum, but rather their *order* (or *rank*) in the sequence.

Definition 12. Let $\mathsf{order} : \mathbb{R}^n \to \mathrm{Perm}(n)$ be the function that computes the order (or rank) of the elements of a vector. For example, $\mathsf{order}([2012, 1, 15]) = [2, 0, 1]$.

Definition 13. The *Spearman correlation* between two sequences x, y is the Pearson correlation of their order vectors: $\mathsf{spear}(x, y) \triangleq \mathsf{corr}(\mathsf{order}(x), \mathsf{order}(y))$.

Lemma 14. *The Spearman correlation detects any nonlinear monotonic relation:* $\mathsf{spear}(x, y) = \pm 1$ *if and only if* $y = g(x)$ *for some monotonic function* g.

We use this fact to check whether there exists a monotonic function f such that $\mathrm{Y}_{ij} = f(d(s_i, s_j))$. Given a solution $\{s_i\}_{i=1}^n$, we compute the corresponding distance matrix, and then compute the Spearman correlation of the distance matrix to the similarity matrix. To that end, we need to first unroll the matrices into a vector using the operator $\mathsf{vec} : \mathbb{R}^{n \times n} \to \mathbb{R}^{n^2}$.

Definition 15. The *Spearman score* of a solution $\{s_i\}_{i=1}^n$ is the Spearman correlation between the (flattened) similarity matrix and the (flattened) distance matrix $\mathbf{D} = [\mathrm{D}_{ij}] = [d(s_i, s_j)]$:

$$\rho_{\text{sp}}(\{s_i\}) \triangleq |\mathsf{spear}(\mathsf{vec}(\mathbf{Y}), \mathsf{vec}(\mathbf{D}))|. \tag{6}$$

The Spearman score is invariant to all symmetries of the problem, including wigglings, which by definition

preserve the ordering of the distances. It is also complete because if $\rho_{\text{sp}}(\{s_i\}, \mathrm{Y}_{ij}) = 1$, then there exists an f such that $\mathrm{Y}_{ij} = f(d(s_i, s_j))$. If the data is corrupted by noise, $\rho_{\text{sp}} = 1$ might not be attainable. In that case, it makes sense to normalize the score by the score of the ground truth.

Definition 16. The *Normalized Spearman score* is

$$\rho^*_{\text{sp}}(\{\overline{s}_i\}, \{s_i\}) \triangleq \frac{\rho_{\text{sp}}(\{s_i\})}{\rho_{\text{sp}}(\{\overline{s}_i\})}. \tag{7}$$

6 Algorithm

We describe an extension of the classic Shepard-Kruskal algorithm (SK) [34], [35], [36], [37] that we call SKv+w (SK variant + warping). The basic idea of SK is to use standard MDS[5] on Y_{ij} to obtain a first guess for $\{s_i\}$. Given this guess, one can obtain a rough estimate $\tilde{f}$ of f; given $\tilde{f}$, one can apply $\tilde{f}^{-1}$ to Y_{ij} to obtain an estimate of the distances D_{ij}; then one solves again for $\{s_i\}$ using MDS. The SK algorithm does not give accurate metric reconstruction. Our goal was to obtain a general algorithm that could work in all corner cases of the observability analysis. The algorithm described here will be shown to be robust across a diverse set of benchmarks on different manifolds, with a vast variation of shapes of f and noise levels. To this end, we extended the SK algorithm in several ways. In the following, some parts are specific to the manifold: we indicate by $\mathsf{MDS}_{\mathcal{M}}$ a generic implementation of MDS on the manifold $\mathcal{M}$, so that $\mathsf{MDS}_{\mathbb{R}^n}$ is the classical Euclidean MDS, and $\mathsf{MDS}_{\mathbb{S}^n}$ is the spherical MDS employed by Grossmann et al..

EM-like iterations (lines 3–7 of Algorithm 1): A straightforward extension is to iterate the alternate estimation of $\{s_i\}$ and f in an EM-like fashion. This modification has also been introduced in other SK variants [37]. This iteration improves the solution, but still does not give metrically accurate solutions.

Choice of first guess for the distance matrix (line 1): Assuming that the similarities have been normalized ($-1 \leq \mathrm{Y}_{ij} \leq 1$), the standard way to obtain an initial guess D^0_{ij} for the distance matrix is to linearly scale the similarities, setting $\mathrm{D}^0_{ij} \propto 1 - \mathrm{Y}_{ij}$. This implies that, given the perturbed similarities $\mathrm{Y}^\star_{ij} = g(\mathrm{Y}_{ij})$ for some monotone function g, the algorithm starts from a different guess and has a different trajectory. However, we expect the same solution because the sufficient statistics $\mathsf{order}(\mathrm{Y}^\star_{ij}) = \mathsf{order}(\mathrm{Y}_{ij})$ is conserved. The fix is to set $\mathrm{D}^0_{ij} \propto \mathsf{order}(\mathrm{Y}_{ij})$ so that the algorithm is invariant to the shape of f.

Multiple initializations (line 1): We observed empirically that multiple initializations are necessary for the case of $\mathbb{S}^m$. In particular, if one scales D^0_{ij} such that $0 \leq \mathrm{D}^0_{ij} \leq \pi$, all solutions generated have diameter $\leq \pi$; if one scales D^0_{ij} such that $0 \leq \mathrm{D}^0_{ij} \leq 2\pi$, all solutions have diameter $\geq \pi$. Extensive tests show that one of the two starting points always allows convergence to the true solution (the other being stuck in a local minimum). In Algorithm 1 this is represented by a manifold-specific function $\mathsf{init}_{\mathcal{M}}$ returning the list of initial guesses for $\mathbf{D}$.

Non-parametric inversion of f (line 5): We have to find some representation for f, of which we do not know the shape, and use this representation to compute f^{-1}. In this kind of scenarios, a common solution is to use a flexible parametric representation for f, such as splines or polynomials. However, parametric fitting is typically not robust to very noisy data. A good solution is to use completely *non-parametric* fitting of f. Suppose we have two sequences $\{x_i\}, \{y_i\}$ which implicitly model a noisy relation $y_i = f(x_i) + \text{noise}$ for some monotone f. Our goal is to estimate the sequence $\{f^{-1}(y_i)\}$. Let $\mathsf{sorted}(\{x_i\})$ be the sorted sequence $\{x_i\}$. Then non-parametric inversion can be obtained by using the order of $\{y_i\}$ to index into the sorted $\{x_i\}$ array[6]:

$$\{f^{-1}(y_i)\} \simeq \mathsf{sorted}(\{x_i\})[\mathsf{order}(\{y_i\})].$$

This is seen in line 5 applied to the distance and similarity matrices, unrolled using the vec operator. Note that because both matrices are symmetric, the flattening and sorting operation can be restricted on only the upper triangular part.

Spearman Score as convergence criterion (line 3): The iterations are stopped when the Spearman score converges. In practice, we observed that after 4–7 iterations the score has negligible improvement for all benchmarks. This score is also used to choose the best solution among multiple initializations (line 8).

Warping recovery phase (lines 9–12): The most important change we introduce is a "warping recovery" phase that changes the qualitative behavior of the algorithm in the case of $\mathbb{S}^m, m \geq 2$. As explained in the observability analysis, in curved spaces the scale of the points distribution is observable. However, the SK algorithm (i.e., lines 3–7 of Algorithm 1) cannot compensate what we call a linear warping (Definition 6); in fact, it is easy to see that if $\mathbf{D}_0$ is a fixed point of the loop, also $\alpha\mathbf{D}_0$, for $\alpha > 0$, is a fixed point. In other words, the "null space" of the Shepard-Kruskal algorithm appears to be the group of linear warpings. Therefore, we implemented a simple algorithm to find the scale that best embeds the data onto the sphere, based on the fact that if $\mathbf{D}$ is a distance matrix for a set of points on $\mathbb{S}^m$, then the cosine matrix $\cos(\mathbf{D})$ must have rank $m+1$. Therefore, to find the optimal scale, we look for the optimal $\alpha > 0$ such that $\cos(\alpha\mathbf{D})$ is closest to a matrix of rank 3. This is implemented in lines 9–12, where the ratio of the $(m+1)$-th and the $(m+2)$-th singular value is chosen as a robust measure of the rank.

While it would be interesting to observe the improvements obtained by each variation to the original

5. Given an $n \times n$ distance matrix $\mathbf{D}$, the best embedding in $\mathbb{R}^m$ can be found by solving for the top m eigenvectors of an $n \times n$ semidefinite positive matrix corresponding to a "normalized" version of $\mathbf{D}$ [42], [37].

6. The square brackets here indicate indexing into the array, as in most programming languages (e.g., Python).

Algorithm 1 The SKv+w embedding algorithm for a generic manifold $\mathcal{M}$.

Input: similarities $\mathbf{Y} \in \mathbb{R}^{n \times n}$; manifold-specific functions: $\mathsf{MDS}_{\mathcal{M}}, \mathsf{distances}_{\mathcal{M}}, \mathsf{init}_{\mathcal{M}}$. **Output:** $\mathbf{S} \in \mathcal{M}^n$.

1 **for** $\mathbf{D}^0$ **in** $\mathsf{init}_{\mathcal{M}}(\mathsf{order}(\mathbf{Y}))$: # Some manifolds need multiple starting points
2 $\mathbf{S}^0 = \mathsf{MDS}_{\mathcal{M}}(\mathbf{D}^0)$ # Compute first guess by MDS
3 **for** $k = 1, 2, \ldots$ **until** s^k converged:
4 $\mathbf{D}^k = \mathsf{distances}_{\mathcal{M}}(\mathbf{S}^{k-1})$ # Compute current distances
5 $\mathbf{D}^k_\star = \mathsf{vec}^{-1}(\mathsf{sorted}(\mathsf{vec}(\mathbf{D}))[\mathsf{order}(\mathsf{vec}(\mathbf{Y}))])$ # Nonparametric fitting and inversion of f.
6 $\mathbf{S}^k = \mathsf{MDS}_{\mathcal{M}}(\mathbf{D}^k_\star)$ # Embed according to the modified distances.
7 $s^k = \mathsf{spearman\_score}(\mathbf{S}^k, \mathbf{Y})$ # Use the Spearman score for checking convergence
8 $\mathbf{S}^\star = \mathbf{S}^{k^\star}$, where $k^\star = \arg\max_k s^k$ # Find best iteration according to the score.
9 **if** $\mathcal{M}$ is $\mathbb{S}^m, m \geq 2$: # Find optimal warping factor to embed in the sphere.
10 $\mathbf{D}^\star = \mathsf{distances}_{\mathcal{M}}(\mathbf{S}^\star)$
11 $\alpha^\star = \arg\min_\alpha \sigma^\alpha_{m+1}/\sigma^\alpha_{m+2}$, where $\{\sigma^\alpha_i\} = \mathsf{singular\_values}(\cos(\alpha \mathbf{D}^\star))$
12 **return** $\mathsf{MDS}_{\mathcal{M}}(\alpha^\star \mathbf{D}^\star)$ # Embed the warped distances
13 **return** $\mathbf{S}^\star$

$\mathcal{M}$-specific initializations: $\mathsf{init}_{\mathbb{R}^m}(\mathbf{oY}) \triangleq \mathbf{oY}$; $\mathsf{init}_{\mathbb{S}^m}(\mathbf{oY}) \triangleq \{\pi \mathbf{oY}/n^2, 2\pi \mathbf{oY}/n^2\}$.

algorithm, for reasons of space we focus only on the impact of the warping recovery phase. We call SKv the SKv+w algorithm *without* the warping recovery phase (i.e., without the lines 9–12).

Algorithm complexity: The dominant cost of SKv+w lies in the truncated SVD decomposition needed for $\mathsf{MDS}_{\mathcal{M}}$ in the inner loop; the exact decomposition takes $O(n^3)$, which is, in practice, in the order of 5 ms for $n = 100$ and 500 ms for $n = 1000$ on current hardware[7]. There exist faster approximations to speed up the MDS step; see, e.g., the various Nystrom approximations [43]. In practice, the entire calibration algorithm takes a few seconds for the datasets discussed later.

7 Camera Calibration Results

We divide the experimental evaluation in two sections. In this first section, we describe the results of the method for camera calibration. In Section 8, we describe a series of experiments with artificial datasets to demonstrate the various corner cases of the observability analysis.

Hardware: We use three different cameras, covering all practical cases for imaging systems: a perspective camera ("FLIP" in the following), a fish-eye camera ("GOPRO"), and an omnidirectional catadioptric camera ("OMNI"). FLIP: The Flip Mino HD is a $100 consumer-level video recorder (Fig 3a). It has a 45° FOV; it has a 3X optical zoom, not used for these logs. GOPRO: The GOPRO is a $300 rugged professional-level fish-eye camera for outdoor use (Fig 4a). The field of view varies slightly between 127° and 170° according to the resolution chosen; for our tests, we chose a resolution corresponding to a 150° field of view. OMNI: We used a custom-made omnidirectional catadioptric camera (Fig. 5a). This is a small, compact system very popular for micro aerial platforms. The camera is created by connecting a perspective camera to a hyperbolic mirror. The resulting field of view is 360° (horizontally) by 100° (vertically). The images have much lower quality than the FLIP and GOPRO (Fig. 4b). Table 2 summarizes the statistics of the three datasets.

7. Tests executed using Numpy 1.5, BLAS compiled with Intel MKL, on a 2.67Ghz Intel Xeon core.

TABLE 2: Dataset statistics

camera	FOV	*fps*	*resolution*	*subsampling*	n	*length*
FLIP	45°	30	1280×720	24×24 grid	1620	57416
GOPRO	150°	30	1280×720	24×24 grid	1620	29646
OMNI	360°	20	640×480	8×8 grid	1470	13131

Manual calibration: We calibrated the cameras using conventional techniques, to have a reference to which to compare our method. We calibrated the FLIP using the Matlab Calibration Toolbox [3], which uses a pinhole model plus second-order distortion models. We calibrated the GOPRO and the OMNI using the OCamCalib calibration toolbox [14], using a fourth-order polynomial for the radial distortion profile [7], [6], [18]. Examples of the calibration images are shown in Fig. 3c and 4c.

Data collection: The environment in which the log is taken influences the spatial statistics of the images. The data logs were taken in a diverse set of environments. The FLIP data was taken outdoors in the Caltech campus. The GOPRO data was taken in the streets of Philadelphia, a typical urban environment. The OMNI data was taken indoors in a private apartment and an office location. Examples of the images collected are shown in Fig. 3b, 4b, 5b; the full videos are available on the website. In all cases, the cameras were held in one hand and waved around "randomly", trying to exercise at least three degrees of freedom (shoulder, elbow, wrist), so that the attitude of the camera was approximately uniformly distributed in $\mathrm{SO}(3)$. We did not establish a more rigorous protocol, as these informal instructions produced good data. Data taken by exercising only one degree of freedom of the arm (e.g., forearm, with the wrist being fixed) did not satisfy the monotonicity assumption.

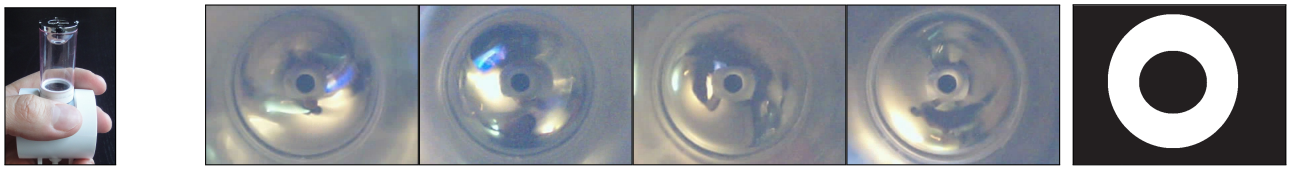
Data processing: For all cameras, the original RGB stream of each pixel was converted to a one-dimensional signal

(a) The Flip Mino **(b)** Some frames from the calibration sequence. **(c)** Calibration patterns

Fig. 3: The FLIP camera is a consumer-level portable video recorder. The data for calibration is taken while walking in the Caltech campus, with the camera in hand, and "randomly" waving the arm, elbow, and wrist.

(a) The GOPRO camera **(b)** Some frames from the calibration sequence **(c)** Calibration pattern

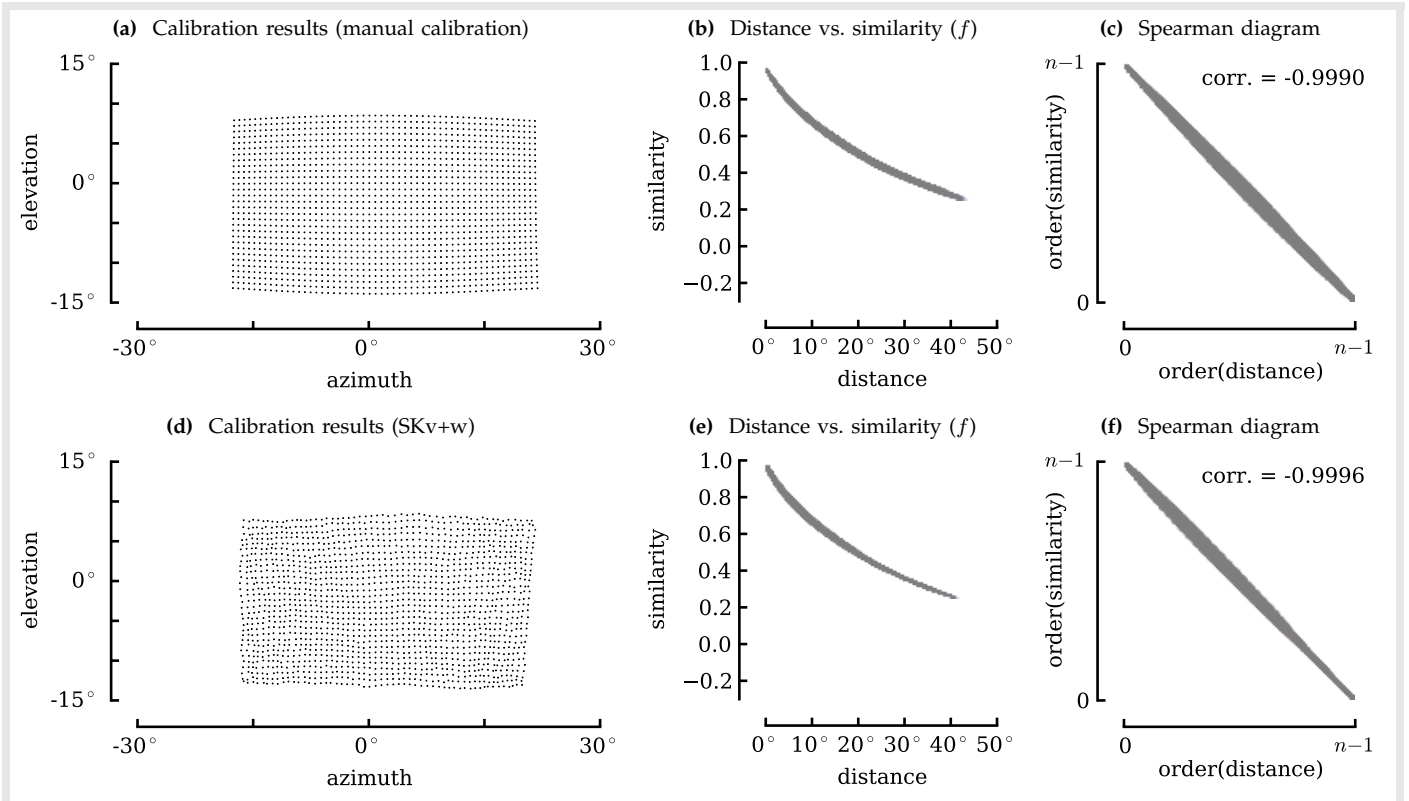
Fig. 4: The GOPRO camera is a rugged consumer camera for outdoors use. It uses a fish-eye lens with 170° field of view.



(a) Catadioptric camera **(b)** Some frames from the calibration sequence **(c)** Mask used

Fig. 5: Note the small dimensions of this omnidirectional catadioptric camera, very well suited for aerial robotics applications. The data quality is much lower than for the FLIP and GOPRO data.

Fig. 6: Calibration results for the FLIP data using $\mathsf{corr}(y)$ as the similarity statistics. See top of page 10 for explanation.



Legend for Figure 6,7,8: The first row (fig. a,b,c) shows the results of calibration using conventional methods, while the second row (d,e,f) shows the results of our algorithm. The first column (a, d) shows the points distribution on the sphere, displayed using azimuth/elevation coordinates. The second column (b, e) shows the joint distribution of pixel distance ($d(s_i, s_j)$) and pixels similarities (Y_{ij}), which, in this case, is the correlation. This is the function f that we should fit. Finally, the third column (c, f) shows order$(d(s_i, s_j))$ vs. order$d(s_i, s_j)$ and their correlation, from which we derive the Spearman score.

Fig. 7: Calibration results for the GOPRO data using corr(y) as the similarity statistics. See above for explanation.

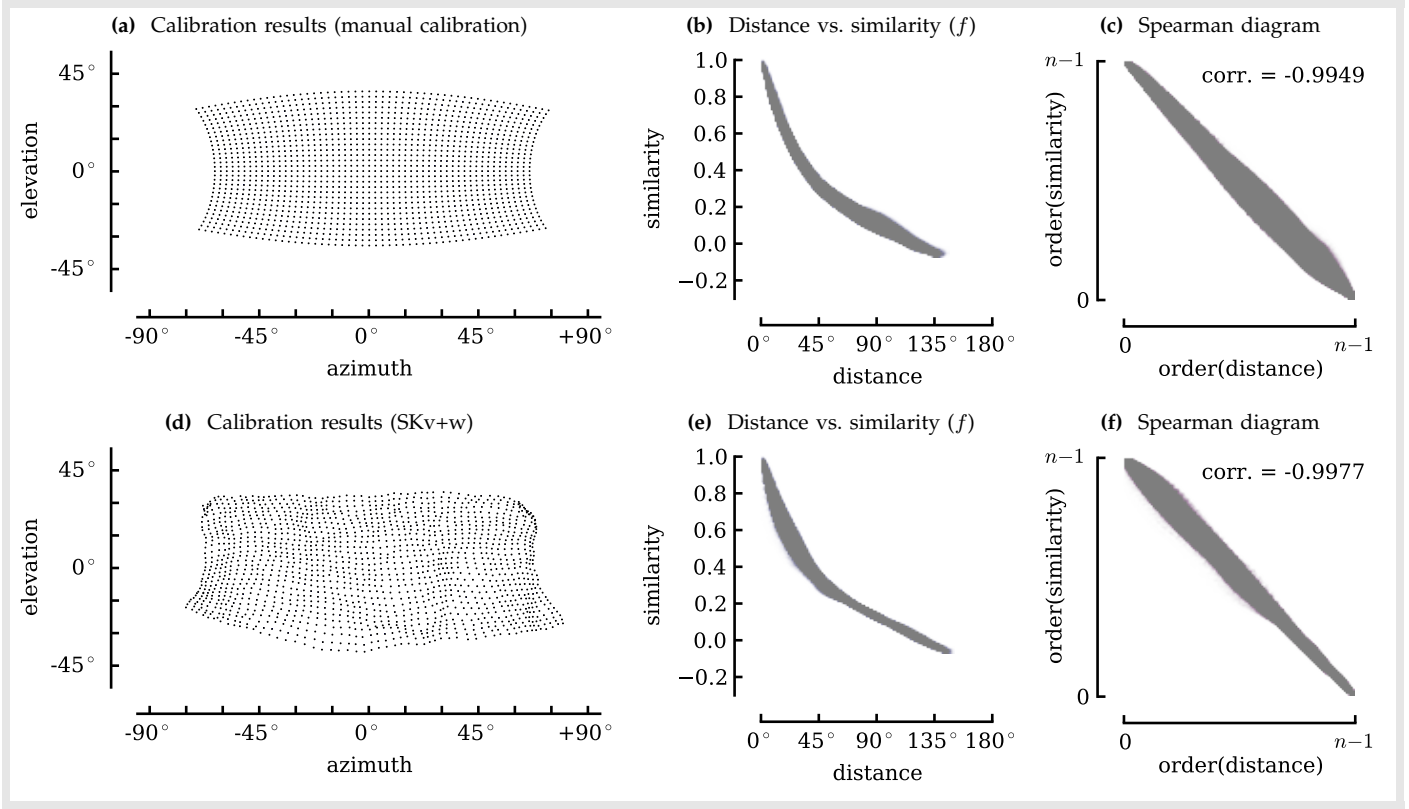
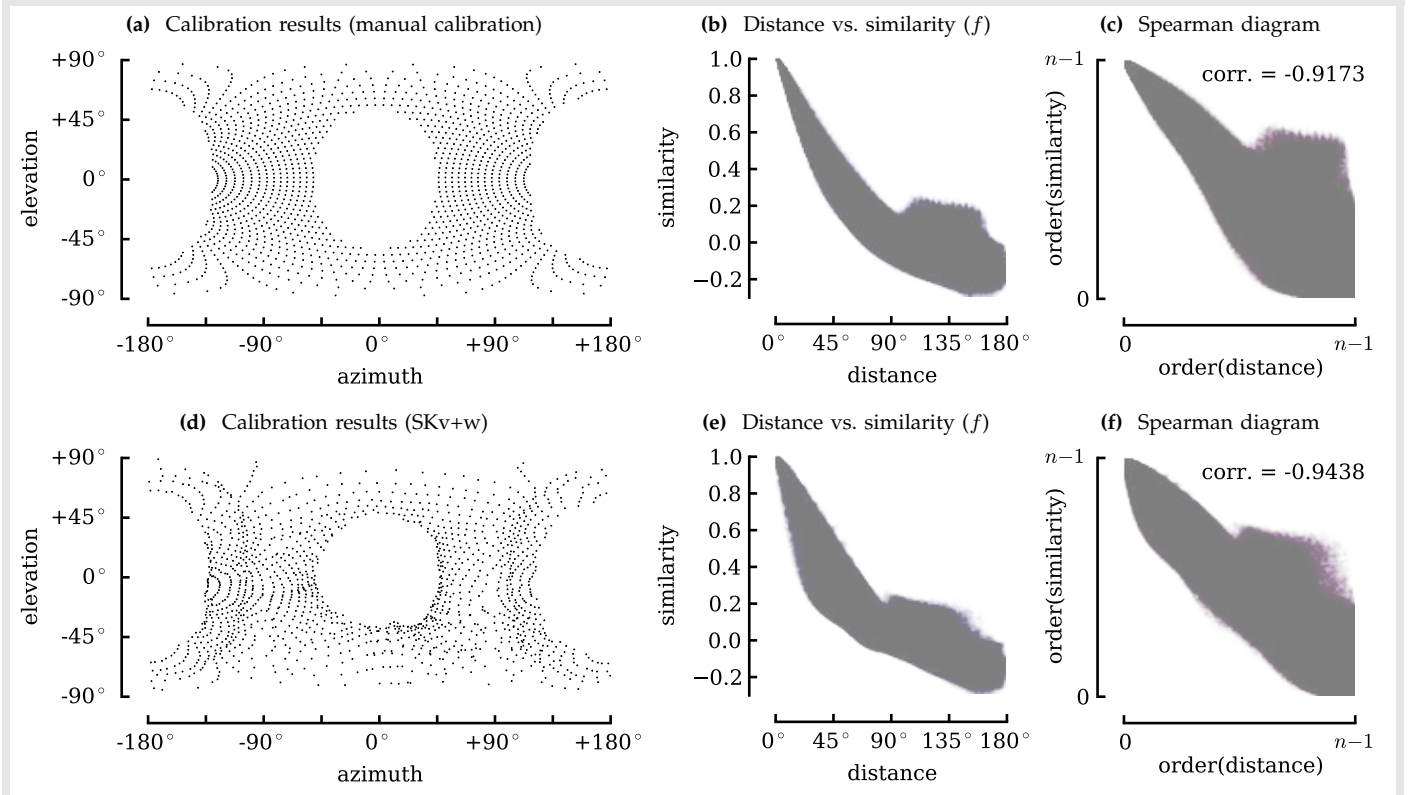


Fig. 8: Calibration results for the OMNI data using corr(y) as the similarity statistics. See the top of this page for explanation.



by computing the luminance. We also sub-sampled the original images with a regular grid so that we could work with a reduced number of points. For the OMNI data, we used masking to only consider the annulus around the center (Fig. 5c), therefore excluding the reflection of the camera in the mirror and the interior of the box which lodged the camera. We used the correlation between the pixel luminance values as the similarity statistics: $\mathrm{Y}_{ij} = \mathrm{corr}(\, y_i(t),\ y_j(t)\,)$, where $y_i(t)$ indicates the luminance of the i-th pixel at time t. This simple statistics gave the best results across the different cameras. Section A-4.1 discusses other possible choices of the similarity statistics.

We found that the monotonicity condition is well verified for all three cameras. To plot these statistics, we assume the calibration results obtained with conventional techniques as the ground truth. The joint distribution of the similarity Y_{ij} and the distance $d(s_i, s_j)$ is shown in Fig. 6b, 7b, 8b. For these logs, the spatial statistics were quite uniform: at a distance of 45°, the inter-pixel correlation was in the range 0.2–0.3 for all three cameras. For the GOPRO and OMNI data, the correlation is 0 at around 90°. The correlation is negative for larger distances. The different average luminance between sky and ground (or floor and ceiling) is a possible explanation for this negative correlation. The OMNI data is very noisy for distances in the range 90°–180°, as the sample correlation converges more slowly for larger distances. To check that the monotonicity condition is satisfied, regardless of the shape of f, it is useful to look at the Spearman diagrams in Fig. 6c, 7c, 8c, for the FLIP, GOPRO, and OMNI, respectively. These diagrams show, instead of similarity (Y_{ij}) versus distance ($d(s_i, s_j)$), the order of the similarities ($\mathrm{order}(\mathrm{Y}_{ij})$) versus the order of the distances ($\mathrm{order}(d(s_i, s_j))$). The correlation of those gives the Spearman score (Definition 15). If there was a perfectly monotonic relation between similarity and distance, the diagram would be a straight line, regardless of the shape of f, and the Spearman score would be 1 (Lemma 14).

7.1 Calibration results

The complete statistics are presented in Appendix B in Table B-1. The results of manual calibration and calibration using our method are graphically shown in Fig. 6d, 7d, 8d. The plots show the data using spherical coordinates (azimuth/elevation).[8] There is a number of intuitive remarks that can be made on the results by direct observations of the resulting point distributions (or, better, its 3D equivalent). For the FLIP data (Fig. 6d) the reconstructed directions lie approximately on a grid, as expected. For this data, and the GOPRO as well, the estimated points are more regular at the center of the field of view than on the borders. This is probably due to the fact that the pixels at the border are less constrained. The estimated FOV is very similar to the result given by the manual calibration (43° instead of 45°). For the GOPRO data (Fig. 7d) the shape of the sensor is well reconstructed, except for the two upper corners of the camera. The estimated FOV matches the manual calibration (153° instead of 150°). For the OMNI data (Fig. 8d) the shape of the sensor is overall well reconstructed, but it is more noisy than the FLIP or GOPRO. This is to be expected as the monotonicity relation is not as well respected (Fig. 8e).

It can be concluded that our method gives results reasonably close to manual calibration, even for cases like the OMNI where the monotonicity condition holds only approximately. As predicted by the observability analysis, the scale can be reconstructed even without knowing anything about the function f.

We now look at quantitative performance measures. As explained before, the only admissible performance measure is the Spearman score (Table 3). When judged by this performance measure, the SKv+w algorithm is slightly better than the manual calibration (the normalized Spearman score is larger than 1). In other words, the estimated distribution is actually a better fit of the similarity data than the manual calibration results. This implies that the imprecision in the estimate is a limitation of the input data rather than of the optimization algorithm; to obtain better results, we should improve on the data rather than improving the algorithm.

The Procrustes error (Equation 3) is the most intuitive performance measure (but not invariant to wiggling). The results are shown in Table 4. The error with respect to manual calibration is an average of 0.7° for the FLIP data, 3.5° for the GOPRO data, and 9.5° for the OMNI data. The table shows both the results with and without the warping phase (SKv+w and SKv, respectively). This makes it clear that the warping phase is necessary to obtain a good estimate of the directions, especially for the FLIP data. The difference is lower for the GOPRO data and negligible for the OMNI data. Intuitively, the warping phase takes advantages of what can be called "second-order" constraints, in the sense that they allow us to

TABLE 3: Calibration results (normalized Spearman score)

dataset			*norm. Spearman score* ρ^*_{sp}			
$\mathcal{S}$	FOV	f	*g. truth*	SKv	SKv+w	MDS
FLIP	45°	corr(y)	1	0.9998	**1.0006**	0.9709
GOPRO	150°	corr(y)	1	1.0027	**1.0029**	0.9702
OMNI	360°	corr(y)	1	**1.0288**	**1.0288**	0.9831

(See complete results in Table B-1b)

TABLE 4: Calibration results (Procrustes error)

dataset			*Procrustes error*		
$\mathcal{S}$	FOV	f	SKv	SKv+w	MDS
FLIP	45°	corr(y)	24.05°	**0.74°**	15.16°
GOPRO	150°	corr(y)	4.72°	**3.53°**	6.20°
OMNI	360°	corr(y)	**9.48°**	**9.48°**	32.43°

(See complete results in Table B-1c)

8. It is challenging to visualize 3D data with 2D projections, because any projection will distort some part of the data. For this reason, we also provide 3D visualization using MATLAB .fig figures, which allows the user to rotate in 3D the data. Click the following links to access the .fig files: mino, GOPRO, OMNI.

establish the scale at small FOV, but they disappear as the FOV tends to zero, because a small enough section of $\mathbb{S}^2$ looks flat (like $\mathbb{R}^2$). Finally, the accuracy of MDS is much lower than SKv or SKv+w. MDS obtains topologically correct solutions, but the scale is never correctly recovered, or the data appears otherwise deformed.

These results outperform those of Grossmann et al. (compare, for example, Figure 13 in [27]), even though their method assumes that the function f is known, obtained through a separate calibration phase. In principle, with much more information, their results should be better. Without having access to their data, we can only speculate on the reason. Perhaps the simplest explanation is that they do not "wave around" the camera for collecting the data; and therefore the monotonicity condition might not be as well satisfied. Moreover, they use a similarity statistics which has very low informative radius (30°).

8 Additional Evaluation

The purpose of this section is to cover all corner cases of the observability analysis in Proposition 8.

Observability class A: The observability class A corresponds to distributions in $\mathbb{S}^2$ where the scale is observable, and it is possible to reconstruct the directions up to wiggling. To illustrate the effect of wiggling, we generated some synthetic datasets, so that the effect of wiggling can be seen independently of the measurement noise. We use as the ground truth the distribution of directions given by manual calibration. We let the function f be $f_{\mathsf{exp}}(d) = \exp(-0.52\,d)$, which is the exponential kernel that best fits the FLIP data (Fig. 6b). The results are shown in Table 5. We can see that the Spearman score for SKv+w is 1, meaning that the solution found is a perfect solution to the problem. However, the Procrustes error is 1.25°. This is the practical demonstration that the Procrustes error is not admissible. This synthetic experiment gives us a sense of what is the accuracy in the directions domain that we can obtain in practice, even if we had perfect measurements of the correlation. The Procrustes error for the real data is in the order of 1°; this means that the contribution of the noise is commensurable to wiggling, and the results would benefit more from using a denser grid rather than longer logs. The Procrustes error is negligible for the omnidirectional data. This shows that an omnidirectional directions distributions makes the problem overall more constrained.

TABLE 5: Benchmarks for class A ($\mathbb{S}^2$) with synthetic data

dataset			*Spearman score*		*Procrustes error*	
$\mathcal{S}$	FOV	f	g. truth	SKv+w	g. truth	SKv+w
FLIP	45°	f_{exp}	**1**	**1.000**	**0°**	1.25°
GOPRO	150°	f_{exp}	**1**	**1.000**	**0°**	0.90°
OMNI	360°	f_{exp}	**1**	**1.000**	**0°**	**0.00°**

(See complete results in Table B-2a and B-2c)

Observability class B: The observability class B corresponds to distributions on $\mathbb{S}^1$ where the scale is observable due to the non-simply connected topology and a function f with large informative radius. For this set of benchmarks we generated the synthetic similarity using three functions f (Table 6). The function $f_{\mathsf{lin}}(d)$ is linear in the distance d. The function $f_{\mathsf{smooth}}(d)$ is nonlinear in the distance, but still invertible on the whole domain. The function f_{steep} is equal to f_{smooth} for $d \in [0, 90°]$ and 0 for $d \geq 90°$. This implies that $\mathsf{infr}(f_{\mathsf{steep}}) = 90°$. We simulated random distributions of points in $\mathbb{S}^1$ with 315° FOV. This satisfies condition (2) because $315°/2 + 90° \geq 180°$. SKv recovers the scale of the distribution in all cases, while MDS recovers it correctly only for linear f (Table 7).

TABLE 6: Kernels used in synthetic benchmarks

	f	$\mathsf{infr}(f)$
$f_{\mathsf{lin}}(d) =$	$0.5 - 0.5d$	180°
$f_{\mathsf{smooth}}(d) =$	$\cos^3(d)$	180°
$f_{\mathsf{steep}}(d) =$	$\max\{\cos^3(d), 0\}$	90°

TABLE 7: Benchmarks for class B ($\mathbb{S}^1$, observable scale)

dataset			*diameter*	
$\mathcal{S}$	FOV	f	MDS	SKv
random dist.	315°	f_{lin}	**318°**	**318°**
random dist.	315°	f_{steep}	127°	**312°**
random dist.	315°	f_{smooth}	114°	**316°**

(See complete results in Table B-4c and B-4a)

Observability class C: The observability class C corresponds to the case of $\mathbb{S}^1$ in which the scale is not observable, because f is not informative enough. We use a mix of synthetic data and real data. For the synthetic data, we simulated a random distribution of directions on $\mathbb{S}^1$ with FOV 45° and 90°. For the real data, we extracted the center scan line of the FLIP and GOPRO data. We know that these pixels lie approximately on a great circle of $\mathbb{S}^2$, therefore this data can be embedded in $\mathbb{S}^1$. Table 8 shows the Spearman score and the Procrustes error obtained by $\mathsf{MDS}_{\mathbb{S}^n}$ and SKv. The most interesting fact about these benchmarks is that they clearly show that, without a proper observability analysis, one might reach erroneous conclusions by considering a non-admissible performance measure. If one were to compare MDS and SKv only on the Procrustes error, one would conclude erroneously that MDS performs better than SKv. However, in reality, SKv gives better results, as can be seen from the Spearman score. What is happening is that, for the non-admissible measure, the intrinsic bias of MDS is better adjusted to the dataset bias.

Observability class D: The observability class D corresponds to the Euclidean case, where the scale is not observable. For this class we used exclusively synthetic data. We simulated a random distribution of points on the $[0, 1] \times [0, 1]$ square, and for the similarities we used the three functions f_{lin}, f_{steep}, f_{smooth}. The data in Table 9 is a simple verification of the fact that MDS provides a correct reconstruction only if similarity is a linear function

TABLE 8: Benchmarks for class C ($\mathbb{S}^1$, unobservable scale)

dataset			*Spearman score*		*Procrustes error*	
$\mathcal{S}$	FOV	f	MDS	SKv	MDS	SKv
FLIP (center)	23°	corr(y)	0.9706	**0.9999**	**13.52°**	25.62°
random dist.	45°	f_{smooth}	0.9987	**0.9999**	**7.47°**	20.08°
GOPRO (center)	75°	corr(y)	0.9592	**0.9988**	**8.58°**	13.01°
random dist.	90°	f_{smooth}	0.9853	**0.9997**	9.22°	**8.41°**

(See complete results in Table B-5c and B-5a)

of the distance, while SKv provides the correct solution regardless of the shape of f.

TABLE 9: Benchmarks for class D ($\mathbb{R}^2$)

dataset		*norm. Spearman score* ρ^*_{sp}		
$\mathcal{S}$	f	g. truth	MDS	SKv
random square	f_{lin}	1	**1.000**	**1.000**
random square	f_{steep}	1	0.7706	**1.000**
random square	f_{smooth}	1	0.9182	**1.000**

(See complete results in Table B-6a)

9 Conclusions

We presented a calibration method that does not need any known calibration pattern, known camera motion, or another calibrated apparatus. It does not have any assumption on the camera model, and is therefore able to calibrate any single-view point cameras. We have shown that calibration-by-correlation is an instance of the problem of metric embedding from non-metric measurements, which appears naturally in many other fields. So far, it has implicitly been assumed that it is not possible to recover the metric information (scale of the distribution) from non-metric measurements. We have given a comprehensive discussion of the observability of the problem, showing that it depends both on the local geometrical characteristics (curvature) as well as the global topological properties (connectedness) of the particular manifold considered. While in Euclidean space it is *never* possible to recover the scale, it *is* possible in spaces of nonzero curvature, or in non-simply connected manifolds such as $\mathbb{S}^1$. We have presented an optimization algorithm based on the classical Shepard-Kruskal algorithm, with several additions that make it robust across manifolds and a variety of benchmarks. The main addition is a "warping recovery phase" that is necessary to obtain the correct scale in the spherical case. In addition to the camera calibration problem, we evaluated the algorithm on a series of synthetic benchmarks, making sure it works as expected in all corner cases of the observability analysis. Therefore, it will likely be useful for problems other than camera calibration that have a similar formalization.

Future work: On the theoretical side, there is still much to do in the analysis of Problem 1. We have provided an observability analysis, but not a result, in the spirit of the Cramér–Rao bound, to establish what noise level can be tolerated on f to obtain a given accuracy. Because the observations can be interpreted as a set of inequalities, while the unknowns live in a continuous space, standard techniques cannot be used. On the algorithmic side, there is still much to do about proving the convergence properties of the algorithm. The algorithm appears to be robust across diverse benchmarks, but we do not have proofs of global convergence. The analysis is difficult because the algorithm is a mix of "continuous" operations (e.g., computing the SVD of a matrix) and "discrete" operations (e.g., reordering the elements of the matrix). Several other extensions are motivated by the problem of camera calibration. As noted before, it is an open question whether one can find a better similarity statistics than the correlation, and whether there exists other motions, other than random rotations, that guarantee a monotonic f. It would also be useful to study extensions of the problem to non-monotonic functions f, or that allow for the statistics to vary in different parts of the field of view. This would allow to use data taken when the camera has a fixed attitude (as in autonomous vehicles). It would also be useful to extend this method to non-central cameras, where each pixel has a direction $s_i \in \mathbb{S}^2$ *and* a spatial position in $\mathbb{R}^3$. Finally, the most immediate future work is integrating this method with parametric methods that use a prior knowledge of the camera model [7], to obtain the best of both worlds.

References

[1] T. A. Clarke and J. G. Fryer, "The Development of Camera Calibration Methods and Models," *The Photogrammetric Record*, vol. 16, no. 91, pp. 51–66, 1998. doi:10.1111/0031-868X.00113.

[2] P. Sturm, S. Ramalingam, J.-P. Tardif, S. Gasparini, and J. Barreto, "Camera models and fundamental concepts used in geometric computer vision," *Foundations and Trends in Computer Graphics and Vision*, vol. 6, no. 1–2, pp. 1–183, 2011. doi:10.1561/0600000023.

[3] J.-Y. Bouguet, "The matlab calibration toolbox.".

[4] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, 2002. doi:10.1109/34.888718.

[5] D. Gennery, "Generalized camera calibration including fish-eye lenses," *Int. J. of Computer Vision*, vol. 68, no. 3, 2006. doi:10.1007/s11263-006-5168-1.

[6] D. Scaramuzza, A. Martinelli, and R. Siegwart, "A flexible technique for accurate omnidirectional camera calibration and structure from motion," in *Proceedings of IEEE International Conference of Vision Systems (ICVS)*, 2006. doi:10.1109/ICVS.2006.3.

[7] D. Scaramuzza, A. Martinelli, and R. Siegwart, "A toolbox for easy calibrating omnidirectional cameras," in *Int. Conf. on Intelligent Robots and Systems*, 2006. doi:10.1109/IROS.2006.282372.

[8] H. Li and R. Hartley, "Plane-based calibration and auto-calibration of a fish-eye camera," in *ACCV 2006* (P. Narayanan, S. Nayar, and H.-Y. Shum, eds.), vol. 3851 of *Lecture Notes in Computer Science*, pp. 21–30, Springer Berlin / Heidelberg, 2006. doi:10.1007/11612032_3.

[9] D. Scaramuzza and R. Siegwart, *Vision Systems: Applications*, ch. A Practical Toolbox for Calibrating Omnidirectional Cameras. inTech, 2007.

[10] C. Mei and P. Rives, "Single view point omnidirectional camera calibration from planar grids," in *Int. Conf. on Robotics and Automation*, (Rome, Italy), pp. 3945–3950, 4 2007. doi:10.1109/ROBOT.2007.364084.

[11] C. Mei, "Omnidirectional camera calibration toolbox for MATLAB.".

[12] J. Kannala and S. S. Brandt, "A generic camera model and calibration method for conventional, wide-angle and fish-eye lenses," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 28, no. 8, pp. 1335–1340, 2006. doi:10.1109/TPAMI.2006.153.

[13] J. Kannala, "Camera calibration toolbox for generic lenses for MATLAB.". .

[14] D. Scaramuzza, "Ocamcalib: Omnidirectional camera calibration toolbox for matlab.". .

[15] J. Barreto, J. Roquette, P. Sturm, and F. Fonseca, "Automatic Camera Calibration Applied to Medical Endoscopy," in *British Machine Vision Conference*, (London, Royaume-Uni), The British Machine Vision Association (BMVA), 2009.

[16] J. Barreto, "Omnidirectional camera calibration toolbox for MATLAB.". .

[17] L. Puig, J. Bermúdez, P. Sturm, and J. J. Guerrero, "Calibration of omnidirectional cameras in practice: A comparison of methods," *Computer Vision and Image Understanding*, vol. 116, no. 1, pp. 120–137, 2012. doi:10.1016/j.cviu.2011.08.003.

[18] M. Rufli, D. Scaramuzza, and R. Siegwart, "Automatic detection of checkerboards on blurred and distorted images,," in *Int. Conf. on Intelligent Robots and Systems*, (Nice, France), pp. 3121–3126, 9 2008. doi:10.1109/IROS.2008.4650703.

[19] F. Kahl and A. Heyden, "Robust self-calibration and euclidean reconstruction via affine approximation," in *Proceedings of the 14th International Conference on Pattern Recognition*, (Washington, DC, USA), pp. 56–, IEEE Computer Society, 1998.

[20] M. K. Chandraker, S. Agarwal, F. Kahl, D. Nistér, and D. J. Kriegman, "Autocalibration via rank-constrained estimation of the absolute quadric," in *Conf. on Computer Vision and Pattern Recognition*, 2007. doi:10.1109/CVPR.2007.383067.

[21] M. Grossberg and S. Nayar, "The Raxel Imaging Model and Ray-Based Calibration," *Int. J. of Computer Vision*, vol. 61, pp. 119–137, 2 2005. doi:10.1023/B:VISI.0000043754.56350.10.

[22] R. I. Hartley, "Self-calibration from multiple views with a rotating camera," in *Proceedings of the third European conference on Computer vision (vol. 1)*, (Secaucus, NJ, USA), pp. 471–478, Springer-Verlag New York, Inc., 1994. doi:10.1007/3-540-57956-7_52.

[23] F. Espuny and J. B. Gil, "Generic self-calibration of central cameras from two "real" rotational flows," in *The 8th Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras*, 2008. .

[24] S. Ramalingam, P. Sturm, and S. K. Lodha, "Generic self-calibration of central cameras," *Computer Vision and Image Understanding*, vol. 114, no. 2, pp. 210 – 219, 2010. doi:10.1016/j.cviu.2009.07.007.

[25] A. Martinelli, D. Scaramuzza, and R. Siegwart, "Automatic self-calibration of a vision system during robot motion," in *Int. Conf. on Robotics and Automation*, (Orlando, FL), pp. 43–48, 5 2006. doi:10.1109/ROBOT.2006.1641159.

[26] G. Antonelli, F. Caccavale, F. Grossi, and A. Marino, "Simultaneous calibration of odometry and camera for a differential drive mobile robot," in *Int. Conf. on Robotics and Automation*, 2010. doi:10.1109/ROBOT.2010.5509954.

[27] E. Grossmann, J. A. Gaspar, and F. Orabona, "Discrete camera calibration from pixel streams," *Computer Vision and Image Understanding*, vol. 114, no. 2, pp. 198 – 209, 2010. doi:10.1016/j.cviu.2009.03.009.

[28] D. Pierce and B. Kuipers, "Map learning with uninterpreted sensors and effectors," *Artificial Intelligence*, vol. 92, no. 1-2, 1997. doi:10.1016/S0004-3702(96)00051-3.

[29] M. Boerlin, T. Delbruck, and K. Eng, "Getting to know your neighbors: unsupervised learning of topography from real-world, event-based input," *Neural computation*, vol. 21, no. 1, 2009. doi:10.1162/neco.2009.06-07-554.

[30] J. Stober, L. Fishgold, and B. Kuipers, "Sensor map discovery for developing robots," in *AAAI Fall Symposium on Manifold Learning and Its Applications*, 2009. .

[31] J. Modayil, "Discovering sensor space: Constructing spatial embeddings that explain sensor correlations," in *Int. Conf. on Development and Learning*, 2010. doi:10.1109/DEVLRN.2010.557885.

[32] J. W. Sammon, "A nonlinear mapping for data structure analysis," *IEEE Transactions on Computers*, vol. 18, pp. 401–409, 5 1969. doi:10.1109/T-C.1969.222678.

[33] R. C. T. Lee, J. R. Slagle, and H. Blum, "A triangulation method for the sequential mapping of points from n-space to two-space," *IEEE Transactions on Computers*, vol. 26, pp. 288–292, 3 1977. doi:10.1109/TC.1977.1674822.

[34] R. Shepard, "The analysis of proximities: Multidimensional scaling with an unknown distance function (part I)," *Psychometrika*, vol. 27, no. 3, pp. 125–140, 1962. doi:10.1007/BF02289630.

[35] R. Shepard, "The analysis of proximities: Multidimensional scaling with an unknown distance function (part II)," *Psychometrika*, vol. 27, no. 3, pp. 219–246, 1962. doi:10.1007/BF02289621.

[36] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonparametric hypothesis," *Psychometrika*, vol. 29, no. 1, pp. 1–27, 1964. doi:10.1007/BF02289565.

[37] S. L. France and J. J. Carroll, "Two-way multidimensional scaling: A review," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 99, pp. 1–18, 2010. doi:10.1109/TSMCC.2010.2078502.

[38] T. Cox and M. Cox, *Multidimensional Scaling*. Boca Raton, FL: Chapman & Hall / CRC, 2001.

[39] Y. Shang, W. Rumi, Y. Zhang, and M. Fromherz, "Localization from connectivity in sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 11, pp. 961–974, 2004. doi:10.1109/TPDS.2004.67.

[40] D. L. Ruderman and W. Bialek, "Statistics of natural images: Scaling in the woods," *Physical Review Letters*, vol. 73, pp. 814–817, Aug. 1994. doi:10.1103/PhysRevLett.73.814.

[41] S. Agarwal, J. Wills, L. Cayton, G. Lanckriet, D. Kriegman, and S. Belongie, "Generalized non-metric multidimensional scaling," in *Eleventh International Conference on Artificial Intelligence and Statistics*, 2007. .

[42] J. C. Gower and G. B. Dijksterhuis, *Procrustes problems*, vol. 30 of *Oxford Statistical Science Series*. Oxford, UK: Oxford University Press, 2004.

[43] J. C. Platt, "FastMap, MetricMap, and Landmark MDS are all Nystrom algorithms," in *In Proceedings of 10th International Workshop on Artificial Intelligence and Statistics*, pp. 261–268, 2005. .

Andrea Censi is a post-doctoral scholar in Computing and Mathematical Sciences at the California Institute of Technology. He received the Laurea and Laurea Specialistica degrees (summa cum laude) in control engineering and robotics from Sapienza University of Rome, Italy, in 2005 and 2007, respectively, and a Ph.D. in Control & Dynamical Systems from the California Institute of Technology in 2012. His research interests include perception, estimation, and learning in natural and artificial systems.

Davide Scaramuzza is Professor of Robotics at the Artificial Intelligence Lab of the University of Zurich, where he leads the Robotics and Perception Group, and Adjunct Faculty at ETH Zurich of the Master in Robotics Systems and Control. He received his PhD (2008) in Robotics and Computer Vision at ETH Zurich. He was Postdoc at both ETH Zurich and the University of Pennsylvania. He led the 2009–2012 *sFly* European project, which focused on autonomous vision-based navigation of micro helicopters in GPS-denied environments. He was awarded the Robotdalen Scientific Awards (2009) and the European Young Researcher Award (2012), sponsored by the IEEE and the European Commission. He is coauthor of the 2nd edition of the book *Introduction to Autonomous Mobile Robots* (MIT Press). He is also author of the first open-source *Omnidirectional Camera Calibration Toolbox* for MATLAB, which, besides thousands of downloads worldwide, is also currently used at NASA, Philips, Bosch, and Daimler. His research interests are field and service robotics, intelligent ground and flying vehicles, and computer vision. Specifically, he investigates the use of cameras as the main sensors for robot navigation, mapping, exploration, reasoning, and interpretation.

Calibration by correlation using metric embedding from non-metric similarities

Supplementary materials

A-1 Definitions

For convenience of the reader, we also repeat various definitions which have been omitted or only stated informally. Introductory references for the differential geometric concepts used in this paper are [1], [2].

$\mathbb{R}^+_\circ$ is the set of positive reals. The generic m-sphere is $\mathbb{S}^m$. $\mathbb{S}^2$ is the 3D sphere, and $\mathbb{S}^1$ is the circle. The target manifold is $\mathcal{M}$ and d indicates its geodesic distance. The set of unknown points to be reconstructed is $\mathcal{S} = \{s_i\}$.

Definition A-1. The *radius* of a set $\mathcal{S}=\{s_i\}$ is defined as

$$\mathsf{rad}(\mathcal{S}) \triangleq \min_i \max_j d(s_i,s_j). \tag{1}$$

The *diameter* is twice the radius.

Definition A-2. The *informative radius* $\mathsf{infr}(f)$ of f is the maximum r such that f is invertible in $[0,r]$.

Definition A-3. An *isometry* is a map $\varphi : \mathcal{M} \to \mathcal{M}$ that preserves distances: $d(\varphi(s_i), \varphi(s_j)) = d(s_i, s_j)$.

Definition A-4. A *conformal map* is a map that preserves the angles between geodesics.

A map is conformal if and only if its Jacobian is proportional to an orthogonal matrix.

Definition A-5. A *generic warping* is a map $\varphi_m : \mathcal{M} \to \mathcal{M}$ such that $d(\varphi(s_1), \varphi(s_2)) = m(d(s_1, s_2))$, for some monotonic function $m : \mathbb{R}^+_\circ \to \mathbb{R}^+_\circ$.

Definition A-6. A *linear warping* is a map φ_α from $\mathcal{M}$ to itself such that $d(\varphi_\alpha(s_1), \varphi_\alpha(s_2)) = \alpha d(s_1, s_2)$ for some $\alpha > 0$.

Definition A-7. A *wiggling* of a set $\{s_i\} \subset \mathcal{M}$ is a map $\varphi : \mathcal{M} \to \mathcal{M}$ that preserves the order of distances: for all i, j, k, l: $d(s_i, s_j) < d(s_k, s_l) \Leftrightarrow d(\varphi(s_i), \varphi(s_j)) < d(\varphi(s_k), \varphi(s_l))$.

Definition A-8. A *geodesic curve* $g(A, B, t)$ from point A to point B, for $t \in [0, 1]$, is the curve on the manifold such that

$$\begin{aligned} d(g(A,B,t),A) &= td(A,B), \\ d(g(A,B,t),B) &= (1-t)d(A,B). \end{aligned}$$

In particular, $g(A, B, 0) = A$ and $g(A, B, 1) = B$.

A-2 Proof of Proposition 2

We use some language (Haar measure) from group theory; for reference, see, e.g., [3], [4].

The luminance at pixel $s \in \mathbb{S}^2$ at time t can be written as

$$y(s,t) = h(\boldsymbol{t}(t), \mathbf{R}(t)\, s),$$

where $\boldsymbol{t} \in \mathbb{R}^m$ is the sensor position, $\mathbf{R} \in \mathsf{SO}(3)$ is the sensor orientation, and $h : \mathbb{R}^3 \times \mathbb{S}^2 \to \mathbb{R}$ is a function that describes the environment. In the following, we drop the dependence on time.

We will show that the statistics depends only on the distance between two pixels, by showing that any pair of pixels with the same distance give the same value of the statistics. Consider two pairs of pixels (s_i, s_j) and (s_k, s_l) having the same distance:

$$d(s_i, s_j) = d(s_k, s_l).$$

Because $d(s_i, s_j) = d(s_k, s_l)$, there exists an $\mathbf{X}$ such that

$$s_k = \mathbf{X} s_i, \qquad s_l = \mathbf{X} s_j. \tag{2}$$

If the probability distribution of $\mathbf{R}$ is uniform on $\mathsf{SO}(3)$, that is, it is the Haar measure $\mathsf{SO}(3)$, then it is also invariant to a rotation (i.e., left/right actions): for all functions z and rotations $\mathbf{X}$, $\mathbb{E}\{z(\mathbf{R})\} = \mathbb{E}\{z(\mathbf{R}\mathbf{X})\}$.

If we choose the $\mathbf{X}$ such that (2), we obtain that

$$\begin{aligned} \mathbb{E}\{g(y(s_i), y(s_j))\} &= \mathbb{E}\{g(h(\boldsymbol{t}, \mathbf{R}\, s_i), h(\boldsymbol{t}, \mathbf{R}\, s_j))\} \\ &= \mathbb{E}\{g(h(\boldsymbol{t}, \mathbf{R}\,\mathbf{X} s_i), h(\boldsymbol{t}, \mathbf{R}\,\mathbf{X} s_j))\} \qquad (3) \\ &= \mathbb{E}\{g(y(s_k), y(s_l))\}. \qquad (4) \end{aligned}$$

A-3 Proof of Proposition 8

A-3.1 Proof overview

The starting point is considering that the largest unobservable transformations are the set of wigglings (Definition 7), because they are exactly those that keep constant the order of the inter-points distances, which is the sufficient statistics for the estimation problem. All other symmetries—isometries (Definition A-3), linear warping (Definition 6), generic warping (Definition A-5) are a specialized version of wigglings. Moreover, an isometry is a linear warping with $\alpha = 1$, and a linear warping is a specialization of a generic warping. In summary, just by the definition of the various transformations, we have the following chain of inclusions:

$$\text{isometries} \subset \begin{matrix}\text{linear}\\ \text{warpings}\end{matrix} \subset \begin{matrix}\text{generic}\\ \text{warpings}\end{matrix} \subset \text{wigglings}.$$

Isometries and warpings are very structured transformations, but wigglings are in general discontinuous. The next step in the analysis is understanding in what cases the set of wigglings is more structured. Proposition A-9 shows that, as the number of points becomes large (in the limit, infinite), wigglings are constrained to be generic warpings. Thus, if $\mathcal{S}$ has an infinite number of points, we have the following:

$$\text{isometries} \subset \begin{matrix}\text{linear}\\ \text{warpings}\end{matrix} \subset \begin{matrix}\text{generic}\\ \text{warpings}\end{matrix} \overset{n\to\infty}{=} \text{wigglings}.$$

With this assumption, we now can study a much more well-behaved set of transformations. Proposition A-10 gives the unexpected result that, in general, there exist no generic nonlinear warpings (Definition A-5), a result that does not depend on the manifold yet (i.e., we did not consider topology or curvature). Intuitively, there is no way to deform the distances in a nonlinear way that maintains the consistency of all constraints. The proof is based on an elementary argument based on the fact that any generic warping must preserve geodesics (Lemma A-11). Thus, only by assuming that the number of points is large, and with no assumptions on the manifold, we can conclude that:

$$\text{isometries} \subset \begin{matrix}\text{linear}\\ \text{warpings}\end{matrix} = \begin{matrix}\text{generic}\\ \text{warpings}\end{matrix} = \text{wigglings}$$

This means that the largest group of symmetries of the problem is composed by linear warping. At this point, we have to consider the property of the manifold. Proposition A-12 shows that, if the manifold has nonpositive curvature, then all linear warpings are necessarily isometries (the scaling factor is 1):

$$\text{isometries} \overset{\mathcal{M}\ \text{curved}}{=} \begin{matrix}\text{linear}\\ \text{warpings}\end{matrix} = \begin{matrix}\text{generic}\\ \text{warpings}\end{matrix} = \text{wigglings}$$

This means that for the sphere $\mathbb{S}^2$ and the hyperbolic plane, isometries are the largest group of symmetries. This is surprising, because it means that *we can recover the scale*, even though the measurements available are completely non-metric. Instead, for Euclidean spaces, it is easy to see that a linear warping is always unobservable. Finally, Proposition A-13 discusses the special case of the circle. Because the topology is not simply connected, it is possible to establish additional constraints: intuitively, an arbitrary warping is not allowed, because if the distribution $\mathcal{S}$ is inflated too much, the tails will "crash" into each other and violate the problem constraints.

A-3.2 Proof details

Proposition A-9. *If $\mathcal{S}$ is a connected open set, all wigglings are generic warpings.*

Proof: The intuition is that a non-trivial wiggling is possible only if there are "gaps" between the points; as the points get denser, the gaps close and the wiggling degenerates to a warping.

Note that the definition of wiggling does not imply any particular property of the map φ such as continuity. It is a map defined only the subset $\mathcal{S}$ of $\mathcal{M}$. There is no information of how φ behaves outside of $\mathcal{S}$. However, if $\mathcal{S}$ is an open subset of $\mathcal{M}$, then necessarily φ must have certain regularities.

First of all, it should necessarily be a continuous map. This can be seen directly from the relation $d(s_i, s_j) < d(s_k, s_l) \Leftrightarrow d(\varphi(s_i), \varphi(s_j)) < d(\varphi(s_k), \varphi(s_l))$ if we let $s_i = s_k$ and consider two sequences $s_j^{(m)} \overset{m\to\infty}{\to} s_i$ and $s_l^{(m)} \overset{m\to\infty}{\to} s_k$.

Consider two pairs of points s_i, s_j at distance $\delta = d(s_i, s_j)$. Consider two other pairs of points s_k, s_l with the same relative distance $\delta = d(s_k, s_l)$ — because the set is open, and the distance is continuous, s_k can be found in a neighborhood of s_i and s_j in a neighborhood of s_l. Because $d(s_i, s_j) = d(s_k, s_l)$, the wiggling direction constraint implies that $d(\varphi(s_i), \varphi(s_j)) = d(\varphi(s_k), \varphi(s_l))$. Because s_k, s_l have no other relation to s_i, s_j other than their distance, it follows that the distance of two points transformed by φ only depends on their initial distance: $d(\varphi(s_i), \varphi(s_j)) = m(d(s_i, s_j))$, for some possibly nonlinear function m. Because φ is continuous, this holds for all points in $\mathcal{S}$, therefore φ is a generic warping. □

Proposition A-10. *All generic warpings are linear warpings.*

Proof: The proof relies on Lemma A-11 below, which says that generic warpings preserve the geodesics. This means that, if the midpoint between A and B is C, then $\varphi(C)$ is the midpoint between $\varphi(A)$ and $\varphi(C)$. Let $d(A, C) = d(C, B) = \ell$. Then $d(\varphi(A), \varphi(C)) = d(\varphi(C), \varphi(B)) = m(\ell)$. We can find two different expressions for $d(\varphi(A), \varphi(B))$:

$$d(\varphi(A), \varphi(B)) = m(d(A, B)) = m(2\ell), \quad \text{and}$$
$$d(\varphi(A), \varphi(B)) = d(\varphi(A), \varphi(C)) + d(\varphi(C), \varphi(B)) = 2m(\ell).$$

It follows that $m(\ell) = \frac{1}{2}m(2\ell)$. Generalize this reasoning to an equal division of the geodesics in k parts, to derive $m(x) = \frac{1}{k}m(kx)$, for all $x > 0$ and integers $k \geq 1$. Take the derivative of both sides with respect to x to obtain $m'(x) = m'(kx)$. For any $y > 0$, let $x = y/k > 0$, and let $k \to \infty$, to obtain $m'(y) = m'(0)$, which implies that m is a linear function. □

Lemma A-11. *A generic warping preserves geodesics. More formally, for $A, B \in M$ and $t \in [0, 1]$, let $g(A, B, t)$ be the geodesic between A and B. If $\varphi : M \to M$ is a warping, then $g(\varphi(A), \varphi(B), t) = \varphi(g(A, B, t))$.*

Proof: As a base case, we prove the statement for the midpoint. Suppose that there exists a geodesic between A and B. Let C be the midpoint between A and B, with $d(A, C) = d(C, B) = L$. Let $a = \varphi(A)$ and $b = \varphi(B)$ be the transformed points. Let $c = g(a, b, \frac{1}{2})$ be the midpoint between a and b, with $d(a, c) = d(c, b) = \ell$. Using some elementary properties of geodesics, we shall derive that $\varphi(C) = c$.

Because c is the midpoint, the shortest path between a and b goes through c:

$$d(a, c) + d(c, b) \leq d(a, x) + d(x, b), \qquad \text{for all } x.$$

Write this for $x = \varphi(C)$:

$$d(a, c) + d(c, b) \leq d(a, \varphi(C)) + d(\varphi(C), b)$$

On the right-hand side, substitute $d(a, \varphi(C)) = d(\varphi(A), \varphi(C)) = m(d(A, C))$, using the definition of warping. Likewise $d(\varphi(C), b) = d(\varphi(C), \varphi(B)) = m(d(C, B))$, giving

$$d(a, c) + d(c, b) \leq m(d(A, C)) + m(d(C, B)).$$

The point c is the midpoint, so let $\ell = d(a, c) = d(c, b)$, and $L = d(A, C) = d(C, B)$. We obtain that $\ell \leq m(L)$.

We can do the same computation with A and B. Because C is the midpoint between A and B, we have that $d(A, C) + d(C, B) \leq d(A, x) + d(x, B)$, for all x. Write it for $x = \varphi^{-1}(c)$ and substitute $A = \varphi^{-1}(a)$ and $B = \varphi^{-1}(b)$ to obtain $2L \leq d(A, \varphi^{-1}(c)) + d(\varphi^{-1}(c), B) = d(\varphi^{-1}(a), \varphi^{-1}(c)) + d(\varphi^{-1}(c), \varphi^{-1}(b)) = m^{-1}(d(a, c)) + m^{-1}(d(c, b)) = 2m^{-1}(\ell)$, which gives us $\ell \geq m(L)$. Together with $\ell \leq m(L)$, we conclude that $\ell = m(L)$. This means that $d(a, \varphi(C)) = d(\varphi(C), b) = \ell$, and hence $\varphi(C)$ is the midpoint between a and b. Because the midpoint is unique, it follows that $c = \varphi(C)$.

We have proved that $g(\varphi(A), \varphi(B), \frac{1}{2}) = \varphi(g(A, B, \frac{1}{2}))$. By dividing the original geodesics, and applying the reasoning above recursively, one can show that $g(\varphi(A), \varphi(B), \frac{a}{2^b}) = \varphi(g(A, B, \frac{a}{2^b}))$ for all integers $b \geq 0$ and $a \leq 2^b$. The set of dyadic rationals $a/2^b$ is dense in $[0, 1]$, and the functions $t \mapsto g(\varphi(A), \varphi(B), t)$ and $t \mapsto \varphi(g(A, B, t))$ are continuous, because they are compositions of continuous functions. If two continuous functions on the same domain X agree on a dense subset of X, the agree on the whole domain. Therefore, it holds that $g(\varphi(A), \varphi(B), t) = \varphi(g(A, B, t))$ for all $t \in [0, 1]$. □

Proposition A-12. *For $\mathbb{S}^m, m \geq 2$ and the hyperbolic plane, all linear warpings are isometries.*

Proof: This is true for all manifolds with nonzero curvature, but the m-sphere and the hyperbolic plane admit an elementary proof based on spherical/hyperbolic geometry. Firstly, note that a linear warping is a conformal map (Definition A-4) as the Jacobian is uniformly α times an orthogonal matrix. Conformal maps preserve angles between geodesics.

Now it is time to recall high-school facts about spherical geometry: the sides of a spherical triangle are uniquely determined by its angles. The same is true for the hyperbolic plane [2].

Consider now three points in $\mathcal{S}$ and the induced spherical/hyperbolic triangle. Under a linear warping, its internal angles are preserved because a linear warping is conformal. Because the angles are preserved, the sides of the triangle are preserved as well, and therefore the distance between points is unchanged. Hence any linear warping is an isometry. □

Proposition A-13. *If* $\mathcal{M} = \mathbb{S}^1$ *and* $\mathsf{rad}(\mathcal{S}) + \mathsf{infr}(f) < 2\pi$, *a linear warping with* $\alpha \leq (2\pi - \mathsf{rad}(\mathcal{S}))/\mathsf{infr}(f)$ *is unobservable.*

Proof: (sketch) This can be verified directly; the upper bound on α ensures that the tails of $\mathcal{S}$ do not overlap in the informative range of f. This result does not hold for $\mathbb{S}^2$, where the geometry of the problem constrains linear warpings to be isometries ($\alpha = 1$). □

A-4 Additional Experiments

A-4.1 Results for different similarity statistics

Proposition 2 ensures that *any* statistics is a function of the pixel distance, but this result is limited in three ways: 1) it is only an asymptotic result, valid as time tends to infinity; 2) it assumes a perfectly uniform attitude distribution; and 3) it does not ensure that the function f is invertible (monotonic). Therefore, it is still an engineering matter to find a statistics which is 1) robust to finite data size; 2) robust to a non-perfectly uniform trajectory; and 3) has a large invertible radius. An exhaustive treatment of this problem is outside the scope of this paper and delegated to future work. Here, we briefly show the results for three other statistics in addition to the luminance correlation. All statistics are defined as the correlation of an instantaneous function of the luminance and can be efficiently computed using streaming methods. The first variant consists in applying an instantaneous contrast transformation $c : y \mapsto y^2$ to the luminance before computing the correlation:

$$\mathrm{Y}_{ij} = \mathsf{corr}(\, c(y_i(t)),\ c(y_j(t))\,) \tag{5}$$

The second statistic is the correlation of the temporal derivative $\dot{y} = \frac{d}{dt}y$ of the luminance:

$$\mathrm{Y}_{ij} = \mathsf{corr}(\, \dot{y}_i(t),\ \dot{y}_j(t)\,) \tag{6}$$

This was inspired by recent developments in neuromorphic hardware [5]. Finally, we consider the correlation of the *sign* of the luminance change, as it is invariant to contrast transformations:

$$\mathrm{Y}_{ij} = \mathsf{corr}(\, \mathsf{sgn}(\dot{y}_i(t)),\ \mathsf{sgn}(\dot{y}_j(t))\,). \tag{7}$$

Table A-1 shows the Spearman score obtained by using these on the OMNI data (the most challenging dataset). We find, in this case, that the contrast-scaled luminance (5) is slightly better than the simple correlation; the solution found is qualitatively similar (Fig. **??**). The two other similarity statistics (6) and (7) have much lower scores; for them, the monotonicity assumption is not well verified: their distributions are not informative for large distances (Fig. **??**, **??**). It is clear that there is a huge design space for similarity statistics. In the end, we did not find any statistic which was better than the simple correlation *uniformly* for all our three data sets. Therefore, we consider this an open research question.

TABLE A-1
Results with different similarity statistics

dataset		*Spearman score*	
$\mathcal{S}$	f	g. truth	SKv+w
OMNI	$\mathsf{corr}(y)$	0.9173	**0.9438**
OMNI	$\mathsf{corr}(c(y))$	0.9212	**0.9465**
OMNI	$\mathsf{corr}(\dot{y})$	0.8550	**0.9211**
OMNI	$\mathsf{corr}(\mathsf{sgn}(\dot{y}))$	0.8739	**0.9077**

(See complete results in Table **??**)

REFERENCES

[1] M. do Carmo, *Riemannian Geometry*. Birkhauser, 1994.
[2] J. Ratcliffe, *Foundations of hyperbolic manifolds*, vol. 149 of *Graduate Texts in Mathematics*. Springer, 2006. doi:10.1007/978-0-387-47322-2.
[3] V. Varadarajan, *Lie Groups, Lie Algebras, and Their Representation*. Springer, 1984.
[4] J. Rotman, *An introduction to the theory of groups*. Springer-Verlag, 1995.
[5] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128 $\times$ 128 120 db 15 μs latency asynchronous temporal contrast vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 43, pp. 566 –576, Feb. 2008. doi:10.1109/JSSC.2007.914337.

Calibration by correlation using metric embedding from non-metric similarities

Appendix B: complete statistics and visualization

TABLE B-1: Observability case A: real **noisy** data

(a) Spearman score — equation 6

	g. truth	$\text{MDS}_{\mathbb{S}^n}$	SKv	SKv+w
FLIP/$\mathsf{corr}(y)$	0.9990	0.9699	0.9989	0.9996
GOPRO/$\mathsf{corr}(y)$	0.9949	0.9652	0.9975	0.9977
OMNI/$\mathsf{corr}(y)$	0.9173	0.9019	0.9438	0.9438

(b) Normalized Spearman score — equation 7

	g. truth	$\text{MDS}_{\mathbb{S}^n}$	SKv	SKv+w
FLIP/$\mathsf{corr}(y)$	1.0000	0.9709	0.9998	1.0006
GOPRO/$\mathsf{corr}(y)$	1.0000	0.9702	1.0027	1.0029
OMNI/$\mathsf{corr}(y)$	1.0000	0.9831	1.0288	1.0288

(c) Procrustes error — equation 3

	g. truth	$\text{MDS}_{\mathbb{S}^n}$	SKv	SKv+w
FLIP/$\mathsf{corr}(y)$	0.00	15.16	24.05	0.74
GOPRO/$\mathsf{corr}(y)$	0.00	6.20	4.72	3.53
OMNI/$\mathsf{corr}(y)$	0.02	32.43	9.48	9.48

(d) Relative error — equation 4

	g. truth	$\text{MDS}_{\mathbb{S}^n}$	SKv	SKv+w
FLIP/$\mathsf{corr}(y)$	0.00	7.14	11.87	0.36
GOPRO/$\mathsf{corr}(y)$	0.00	1.72	0.95	0.79
OMNI/$\mathsf{corr}(y)$	0.00	2.20	1.80	1.80

(e) Scaled relative error — equation 5

	g. truth	$\text{MDS}_{\mathbb{S}^n}$	SKv	SKv+w
FLIP/$\mathsf{corr}(y)$	0.00	1.79	0.40	0.33
GOPRO/$\mathsf{corr}(y)$	0.00	5.99	2.51	2.55
OMNI/$\mathsf{corr}(y)$	0.00	11.90	8.91	8.91

(f) Diameter — equation 1

	g. truth	$\text{MDS}_{\mathbb{S}^n}$	SKv	SKv+w
FLIP/$\mathsf{corr}(y)$	45.13	75.15	133.19	42.63
GOPRO/$\mathsf{corr}(y)$	150.99	111.82	129.79	153.06
OMNI/$\mathsf{corr}(y)$	329.54	221.53	312.77	312.77

TABLE B-2: Observability case A: synthetic **noiseless** data

(a) Spearman score — equation 6

	g. truth	$\text{MDS}_{\mathbb{S}^n}$	SKv	SKv+w
FLIP/f_{exp}	1.0000	0.9704	0.9992	1.0000
GOPRO/f_{exp}	1.0000	0.9499	0.9999	1.0000
OMNI/f_{exp}	1.0000	0.9739	1.0000	1.0000

(b) Normalized Spearman score — equation 7

	g. truth	$\text{MDS}_{\mathbb{S}^n}$	SKv	SKv+w
FLIP/f_{exp}	1.0000	0.9704	0.9992	1.0000
GOPRO/f_{exp}	1.0000	0.9499	0.9999	1.0000
OMNI/f_{exp}	1.0000	0.9739	1.0000	1.0000

(c) Procrustes error — equation 3

	g. truth	$\text{MDS}_{\mathbb{S}^n}$	SKv	SKv+w
FLIP/f_{exp}	0.00	16.17	24.07	1.25
GOPRO/f_{exp}	0.00	8.04	3.52	0.90
OMNI/f_{exp}	0.02	37.07	0.00	0.00

(d) Relative error — equation 4

	g. truth	$\text{MDS}_{\mathbb{S}^n}$	SKv	SKv+w
FLIP/f_{exp}	0.00	7.54	11.79	0.63
GOPRO/f_{exp}	0.00	2.00	0.55	0.12
OMNI/f_{exp}	0.00	2.95	0.00	0.00

(e) Scaled relative error — equation 5

	g. truth	$\text{MDS}_{\mathbb{S}^n}$	SKv	SKv+w
FLIP/f_{exp}	0.00	1.83	0.26	0.06
GOPRO/f_{exp}	0.00	7.60	0.26	0.12
OMNI/f_{exp}	0.00	9.54	0.00	0.00

(f) Diameter — equation 1

	g. truth	$\text{MDS}_{\mathbb{S}^n}$	SKv	SKv+w
FLIP/f_{exp}	45.13	78.76	134.75	41.13
GOPRO/f_{exp}	150.99	112.33	136.53	154.00
OMNI/f_{exp}	329.54	189.61	329.54	329.54

TABLE B-3: Observability case A: multiple statistics

(a) Spearman score — equation 6

	g. truth	SKv+w
OMNI/corr(y)	0.9173	0.9438
OMNI/corr$(c(y))$	0.9212	0.9465
OMNI/corr$(\dot{y})$	0.8550	0.9211
OMNI/corr$(\mathsf{sgn}(\dot{y}))$	0.8739	0.9077

(b) Normalized Spearman score — equation 7

	g. truth	SKv+w
OMNI/corr(y)	1.0000	1.0288
OMNI/corr$(c(y))$	1.0000	1.0274
OMNI/corr$(\dot{y})$	1.0000	1.0773
OMNI/corr$(\mathsf{sgn}(\dot{y}))$	1.0000	1.0387

(c) Procrustes error — equation 3

	g. truth	SKv+w
OMNI/corr(y)	0.02	9.48
OMNI/corr$(c(y))$	0.02	9.30
OMNI/corr$(\dot{y})$	0.02	16.07
OMNI/corr$(\mathsf{sgn}(\dot{y}))$	0.02	10.24

(d) Relative error — equation 4

	g. truth	SKv+w
OMNI/corr(y)	0.00	1.80
OMNI/corr$(c(y))$	0.00	1.75
OMNI/corr$(\dot{y})$	0.00	2.35
OMNI/corr$(\mathsf{sgn}(\dot{y}))$	0.00	2.61

(e) Scaled relative error — equation 5

	g. truth	SKv+w
OMNI/corr(y)	0.00	8.91
OMNI/corr$(c(y))$	0.00	8.72
OMNI/corr$(\dot{y})$	0.00	14.50
OMNI/corr$(\mathsf{sgn}(\dot{y}))$	0.00	9.22

(f) Diameter — equation 1

	g. truth	SKv+w
OMNI/corr(y)	329.54	312.77
OMNI/corr$(c(y))$	329.54	317.39
OMNI/corr$(\dot{y})$	329.54	284.65
OMNI/corr$(\mathsf{sgn}(\dot{y}))$	329.54	286.48

TABLE B-4: Benchmarks for observability case B (data in $\mathbb{S}^1$, observable scale)

(a) Spearman score — equation 6

	g. truth	$\text{MDS}_{\mathbb{S}^n}$	SKv	SKv+w
$\mathbb{S}^1$, 315deg, f_{steep}	0.8917	0.7739	0.8892	0.8892
$\mathbb{S}^1$, 315deg, f_{smooth}	0.9741	0.8357	0.9740	0.9740
$\mathbb{S}^1$, 315deg, f_{lin}	0.9993	0.8391	0.9993	0.9993

(b) Normalized Spearman score — equation 7

	g. truth	$\text{MDS}_{\mathbb{S}^n}$	SKv	SKv+w
$\mathbb{S}^1$, 315deg, f_{steep}	1.0000	0.8679	0.9972	0.9972
$\mathbb{S}^1$, 315deg, f_{smooth}	1.0000	0.8579	0.9999	0.9999
$\mathbb{S}^1$, 315deg, f_{lin}	1.0000	0.8397	1.0000	1.0000

(c) Procrustes error — equation 3

	g. truth	$\text{MDS}_{\mathbb{S}^n}$	SKv	SKv+w
$\mathbb{S}^1$, 315deg, f_{steep}	0.00	31.24	3.84	3.84
$\mathbb{S}^1$, 315deg, f_{smooth}	0.00	33.46	0.78	0.78
$\mathbb{S}^1$, 315deg, f_{lin}	0.00	40.26	0.07	0.07

(d) Relative error — equation 4

	g. truth	$\text{MDS}_{\mathbb{S}^n}$	SKv	SKv+w
$\mathbb{S}^1$, 315deg, f_{steep}	0.00	2.56	1.58	1.58
$\mathbb{S}^1$, 315deg, f_{smooth}	0.00	2.71	0.56	0.56
$\mathbb{S}^1$, 315deg, f_{lin}	0.00	2.85	0.09	0.09

(e) Scaled relative error — equation 5

	g. truth	$\text{MDS}_{\mathbb{S}^n}$	SKv	SKv+w
$\mathbb{S}^1$, 315deg, f_{steep}	0.00	26.98	5.17	5.17
$\mathbb{S}^1$, 315deg, f_{smooth}	0.00	26.65	1.07	1.07
$\mathbb{S}^1$, 315deg, f_{lin}	0.00	27.63	0.10	0.10

(f) Diameter — equation 1

	g. truth	$\text{MDS}_{\mathbb{S}^n}$	SKv	SKv+w
$\mathbb{S}^1$, 315deg, f_{steep}	317.92	126.81	312.15	312.15
$\mathbb{S}^1$, 315deg, f_{smooth}	317.92	114.04	316.75	316.75
$\mathbb{S}^1$, 315deg, f_{lin}	317.92	95.33	318.10	318.10

(g) Angular correlation

	g. truth	$\text{MDS}_{\mathbb{S}^n}$	SKv	SKv+w
$\mathbb{S}^1$, 315deg, f_{steep}	1.0000	0.9646	0.9991	0.9991
$\mathbb{S}^1$, 315deg, f_{smooth}	1.0000	0.9620	1.0000	1.0000
$\mathbb{S}^1$, 315deg, f_{lin}	1.0000	0.9417	1.0000	1.0000

TABLE B-5: Benchmarks for observability case C (data in $\mathbb{S}^1$, **unobservable** scale)

(a) Spearman score — equation 6

	g. truth	$\text{MDS}_{\mathbb{S}^n}$	SKv
$\mathbb{S}^1$, 90deg, f_{lin}	0.9999	0.9996	0.9999
$\mathbb{S}^1$, 45deg, f_{lin}	0.9999	0.9999	0.9999
$\mathbb{S}^1$, 45deg, f_{smooth}	0.9999	0.9987	0.9999
$\mathbb{S}^1$, 90deg, f_{smooth}	0.9999	0.9853	0.9997
FLIP (center)/$\text{corr}(y)$	0.9997	0.9706	0.9999
GOPRO (center)/$\text{corr}(y)$	0.9996	0.9592	0.9988

(b) Normalized Spearman score — equation 7

	g. truth	$\text{MDS}_{\mathbb{S}^n}$	SKv
$\mathbb{S}^1$, 90deg, f_{lin}	1.0000	0.9997	1.0000
$\mathbb{S}^1$, 45deg, f_{lin}	1.0000	1.0000	1.0000
$\mathbb{S}^1$, 45deg, f_{smooth}	1.0000	0.9988	1.0000
$\mathbb{S}^1$, 90deg, f_{smooth}	1.0000	0.9854	0.9998
FLIP (center)/$\text{corr}(y)$	1.0000	0.9708	1.0002
GOPRO (center)/$\text{corr}(y)$	1.0000	0.9596	0.9992

(c) Procrustes error — equation 3

	g. truth	$\text{MDS}_{\mathbb{S}^n}$	SKv
$\mathbb{S}^1$, 90deg, f_{lin}	0.00	7.27	8.43
$\mathbb{S}^1$, 45deg, f_{lin}	0.00	3.47	20.08
$\mathbb{S}^1$, 45deg, f_{smooth}	0.00	7.47	20.08
$\mathbb{S}^1$, 90deg, f_{smooth}	0.00	9.22	8.41
FLIP (center)/$\text{corr}(y)$	0.00	13.52	25.62
GOPRO (center)/$\text{corr}(y)$	0.00	8.58	13.01

(d) Relative error — equation 4

	g. truth	$\text{MDS}_{\mathbb{S}^n}$	SKv
$\mathbb{S}^1$, 90deg, f_{lin}	0.00	1.56	1.69
$\mathbb{S}^1$, 45deg, f_{lin}	0.00	1.42	8.07
$\mathbb{S}^1$, 45deg, f_{smooth}	0.00	2.79	8.07
$\mathbb{S}^1$, 90deg, f_{smooth}	0.00	1.69	1.62
FLIP (center)/$\text{corr}(y)$	0.00	9.49	19.88
GOPRO (center)/$\text{corr}(y)$	0.00	2.93	3.50

(e) Scaled relative error — equation 5

	g. truth	$\text{MDS}_{\mathbb{S}^n}$	SKv
$\mathbb{S}^1$, 90deg, f_{lin}	0.00	0.45	0.12
$\mathbb{S}^1$, 45deg, f_{lin}	0.00	0.07	0.06
$\mathbb{S}^1$, 45deg, f_{smooth}	0.00	0.43	0.06
$\mathbb{S}^1$, 90deg, f_{smooth}	0.00	3.19	0.30
FLIP (center)/$\text{corr}(y)$	0.00	1.13	0.11
GOPRO (center)/$\text{corr}(y)$	0.00	4.30	0.64

(f) Diameter — equation 1

	g. truth	$\text{MDS}_{\mathbb{S}^n}$	SKv
$\mathbb{S}^1$, 90deg, f_{lin}	89.76	60.27	122.05
$\mathbb{S}^1$, 45deg, f_{lin}	44.88	31.42	122.05
$\mathbb{S}^1$, 45deg, f_{smooth}	44.88	70.33	122.05
$\mathbb{S}^1$, 90deg, f_{smooth}	89.76	106.25	119.91
FLIP (center)/$\text{corr}(y)$	23.12	58.04	125.50
GOPRO (center)/$\text{corr}(y)$	73.41	78.51	128.97

(g) Angular correlation

	g. truth	$\text{MDS}_{\mathbb{S}^n}$	SKv
$\mathbb{S}^1$, 90deg, f_{lin}	1.0000	0.9985	0.9976
$\mathbb{S}^1$, 45deg, f_{lin}	1.0000	1.0000	1.0000
$\mathbb{S}^1$, 45deg, f_{smooth}	1.0000	0.9996	1.0000
$\mathbb{S}^1$, 90deg, f_{smooth}	1.0000	0.9971	0.9976
FLIP (center)/$\text{corr}(y)$	nan	0.9894	0.9999
GOPRO (center)/$\text{corr}(y)$	nan	0.9847	0.9996

TABLE B-6: Benchmarks for observability case D (Euclidean case, unobservable scale)

(a) Spearman score — equation 6

	g. truth	$\mathsf{MDS}_{\mathbb{R}^n}$	SKv
$\mathbb{R}^2$, f_{lin}	0.9998	0.9998	0.9998
$\mathbb{R}^2$, f_{steep}	0.9998	0.7704	0.9998
$\mathbb{R}^2$, f_{smooth}	0.9998	0.9180	0.9998

(b) Normalized Spearman score — equation 7

	g. truth	$\mathsf{MDS}_{\mathbb{R}^n}$	SKv
$\mathbb{R}^2$, f_{lin}	1.0000	1.0000	1.0000
$\mathbb{R}^2$, f_{steep}	1.0000	0.7706	1.0000
$\mathbb{R}^2$, f_{smooth}	1.0000	0.9182	1.0000

(c) Procrustes error — equation 3

	g. truth	$\mathsf{MDS}_{\mathbb{R}^n}$	SKv
$\mathbb{R}^2$, f_{lin}	0.00	0.46	0.03
$\mathbb{R}^2$, f_{steep}	0.00	0.28	0.03
$\mathbb{R}^2$, f_{smooth}	0.00	0.22	0.03

(d) Relative error — equation 4

	g. truth	$\mathsf{MDS}_{\mathbb{R}^n}$	SKv
$\mathbb{R}^2$, f_{lin}	0.0000	0.6295	0.0411
$\mathbb{R}^2$, f_{steep}	0.0000	0.3859	0.0411
$\mathbb{R}^2$, f_{smooth}	0.0000	0.3255	0.0411

(e) Scaled relative error — equation 5

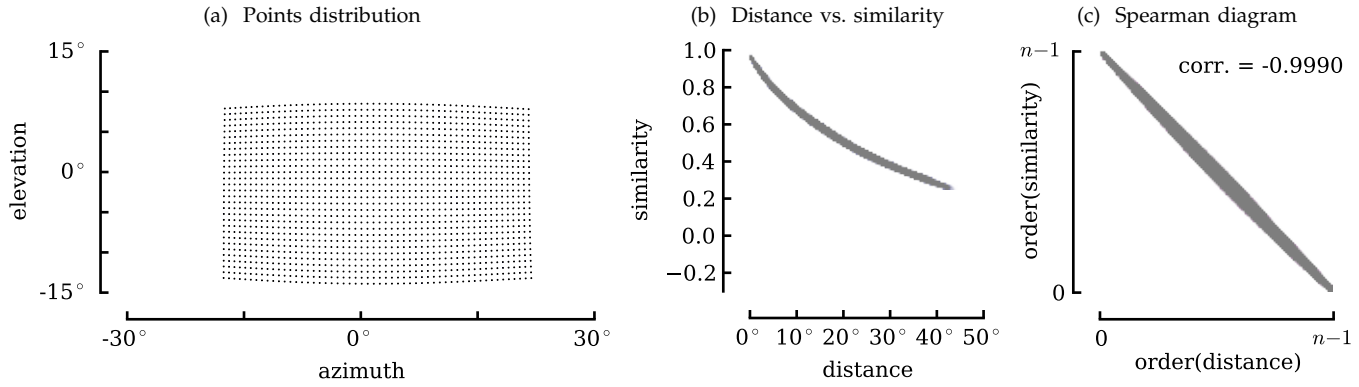
	g. truth	$\mathsf{MDS}_{\mathbb{R}^n}$	SKv
$\mathbb{R}^2$, f_{lin}	0.000	0.003	0.003
$\mathbb{R}^2$, f_{steep}	0.000	0.237	0.003
$\mathbb{R}^2$, f_{smooth}	0.000	0.153	0.003

(f) Diameter — equation 1

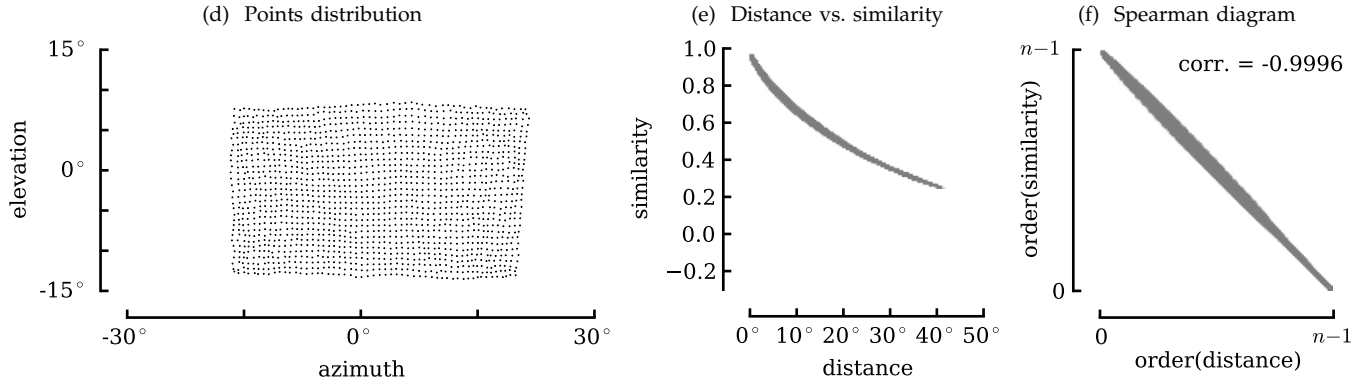
	g. truth	$\mathsf{MDS}_{\mathbb{R}^n}$	SKv
$\mathbb{R}^2$, f_{lin}	2.5302	0.8010	2.6160
$\mathbb{R}^2$, f_{steep}	2.5302	1.1413	2.6160
$\mathbb{R}^2$, f_{smooth}	2.5302	1.1590	2.6160

Fig. B-1: FLIP/corr(y)

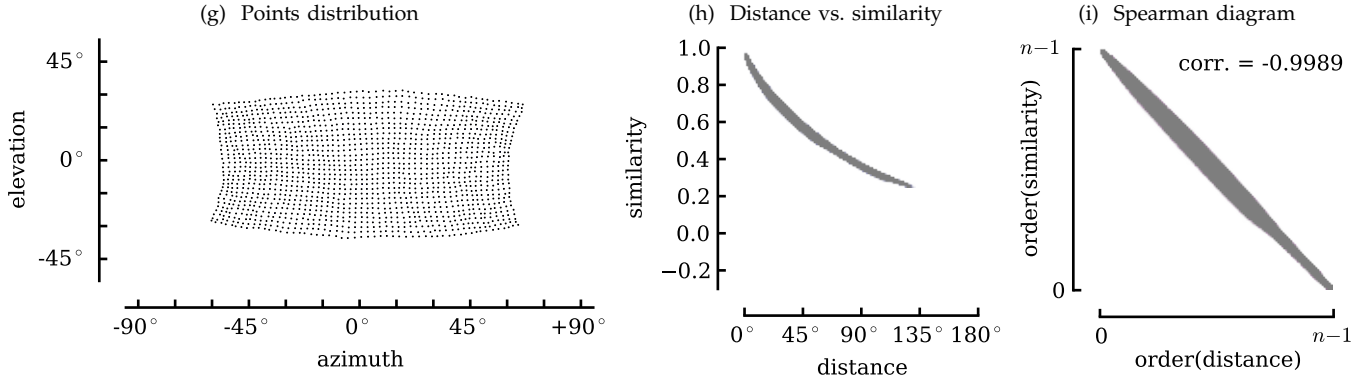
Ground truth



SKv+w



SKv



$\text{MDS}_{\mathbb{S}^n}$

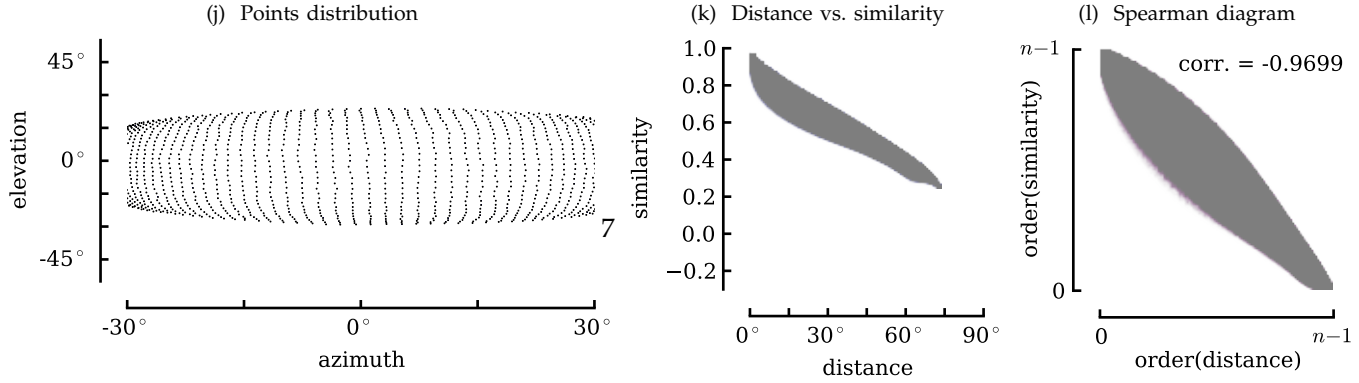
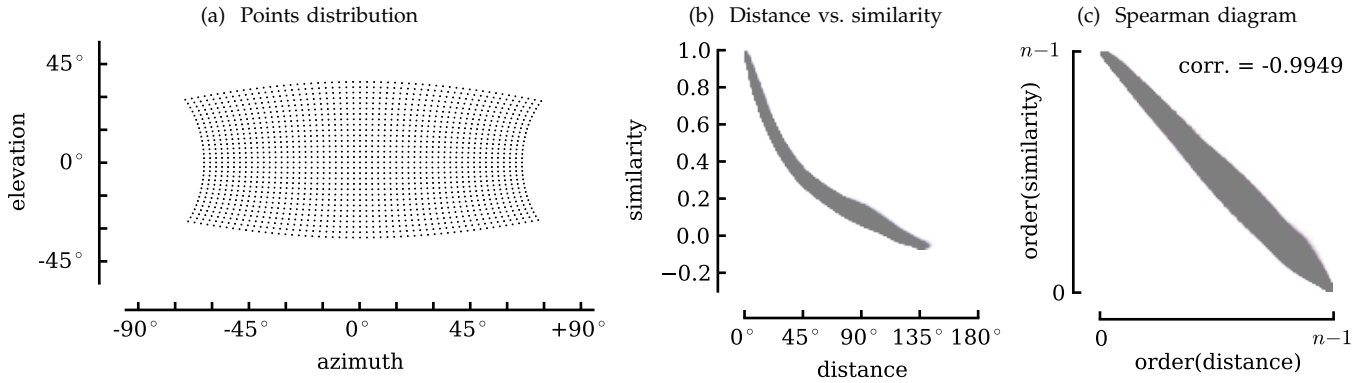
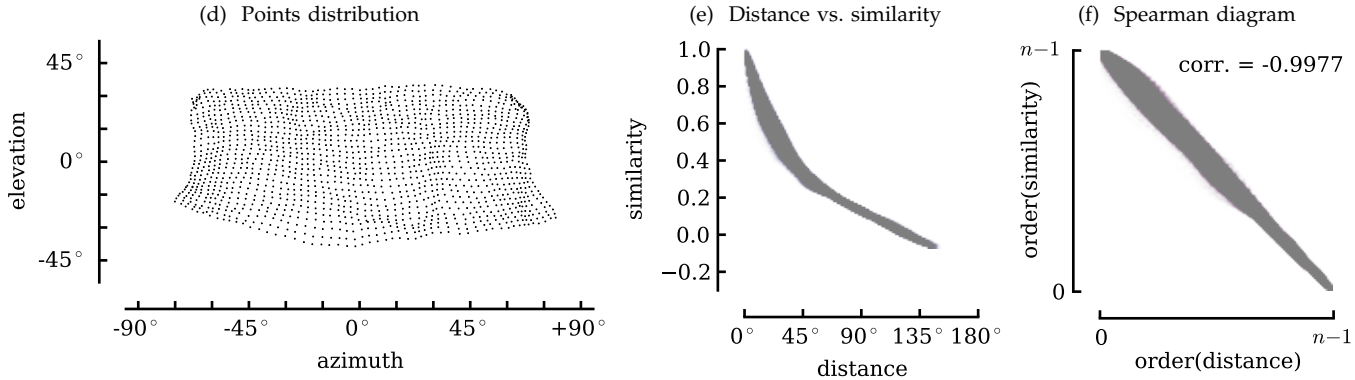


Fig. B-2: GOPRO/corr(y)

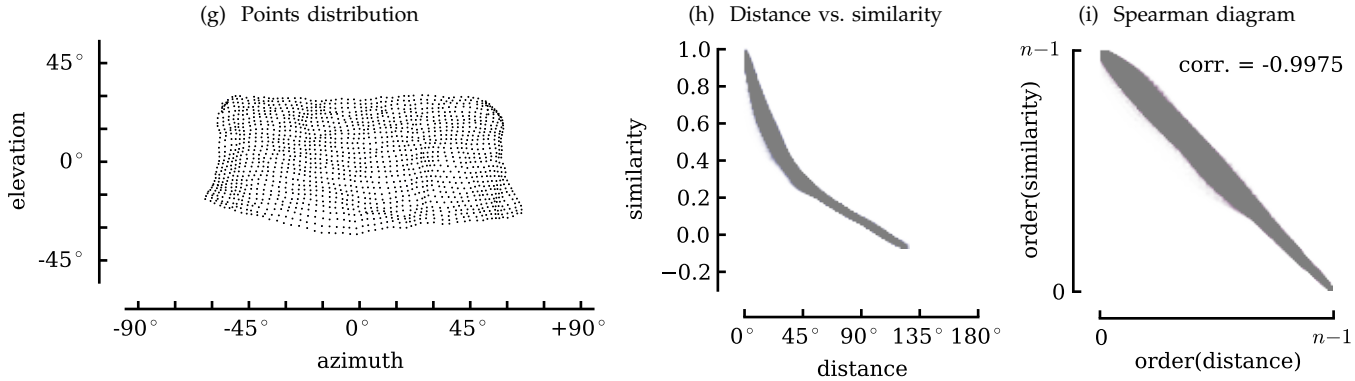
Ground truth



SKv+w



SKv



$\mathsf{MDS}_{\mathbb{S}^n}$

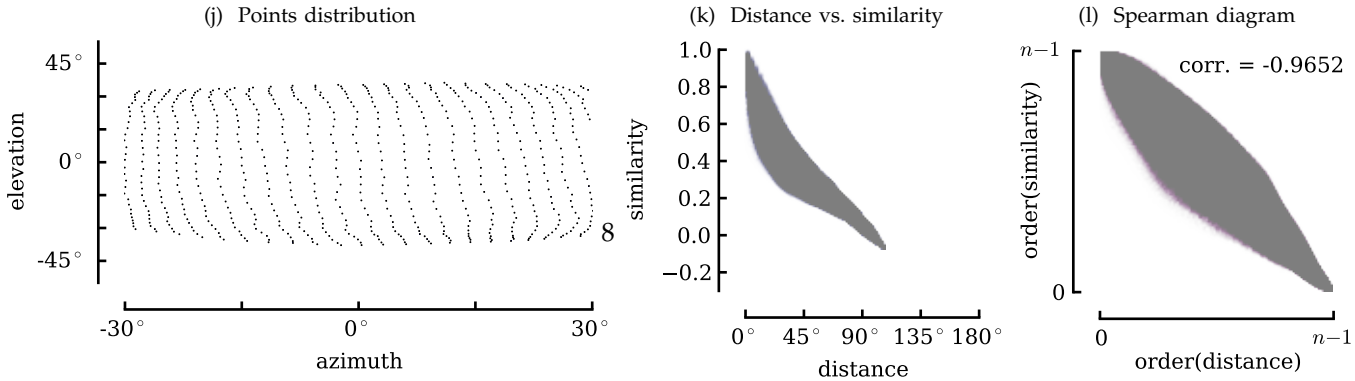
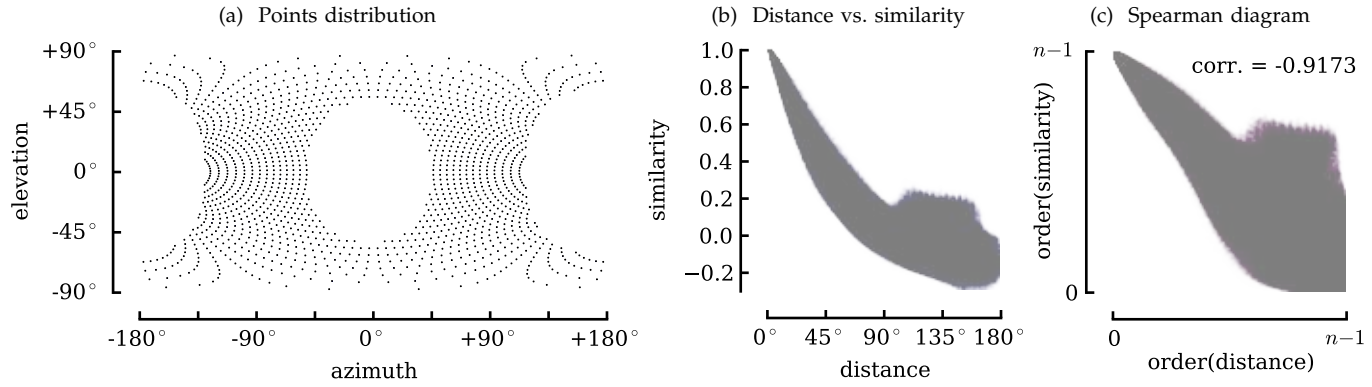
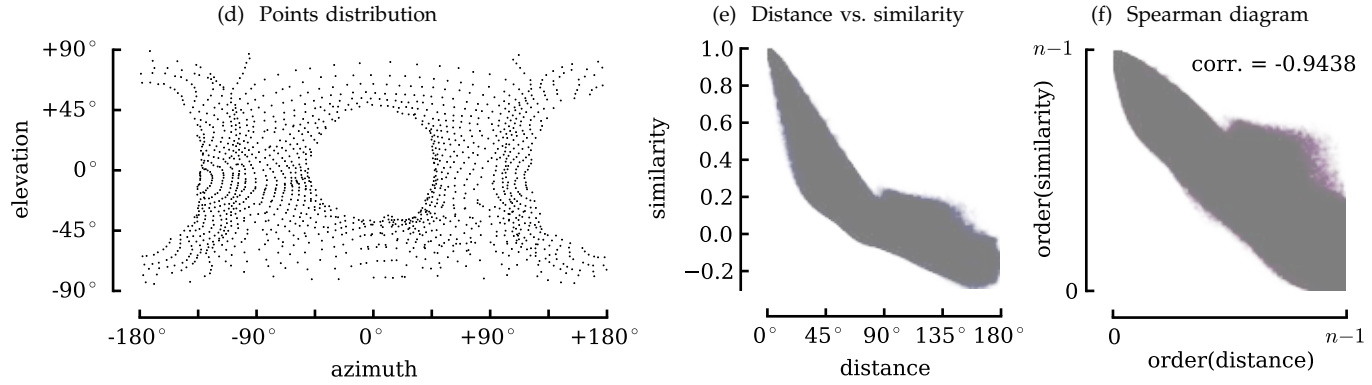


Fig. B-3: OMNI/corr(y)

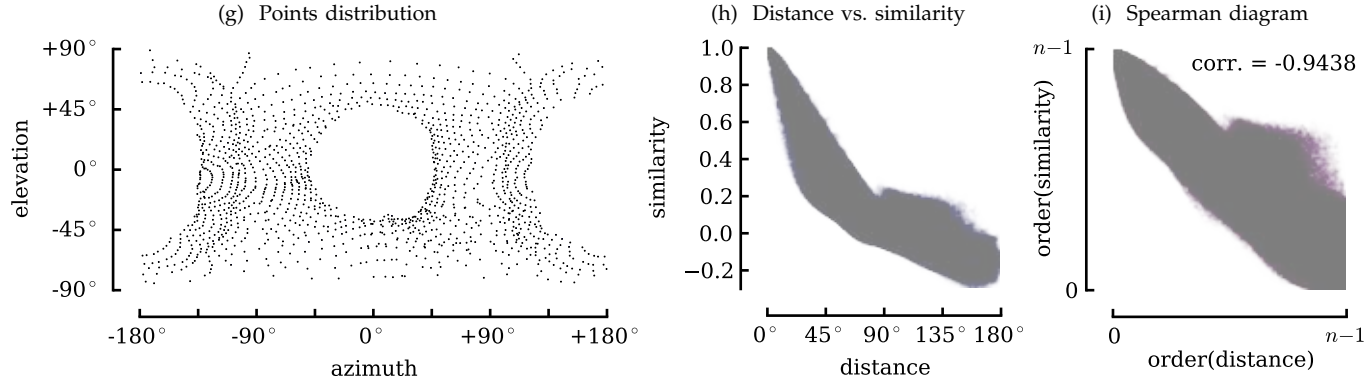
Ground truth



SKv+w



SKv



$\text{MDS}_{\mathbb{S}^n}$

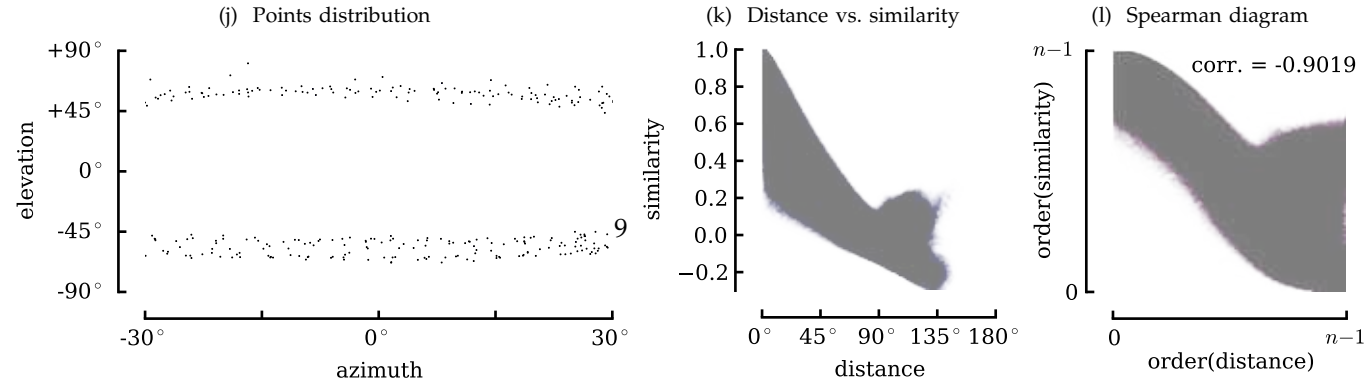
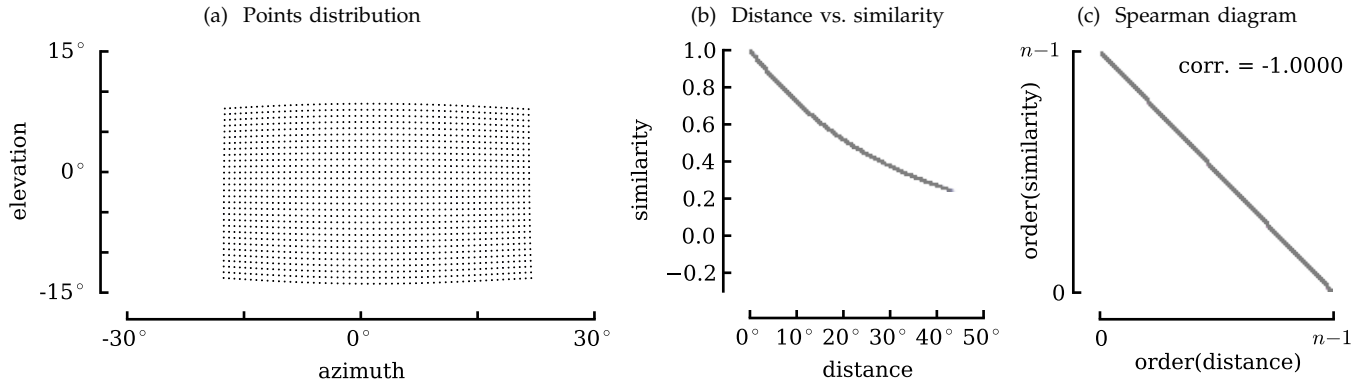
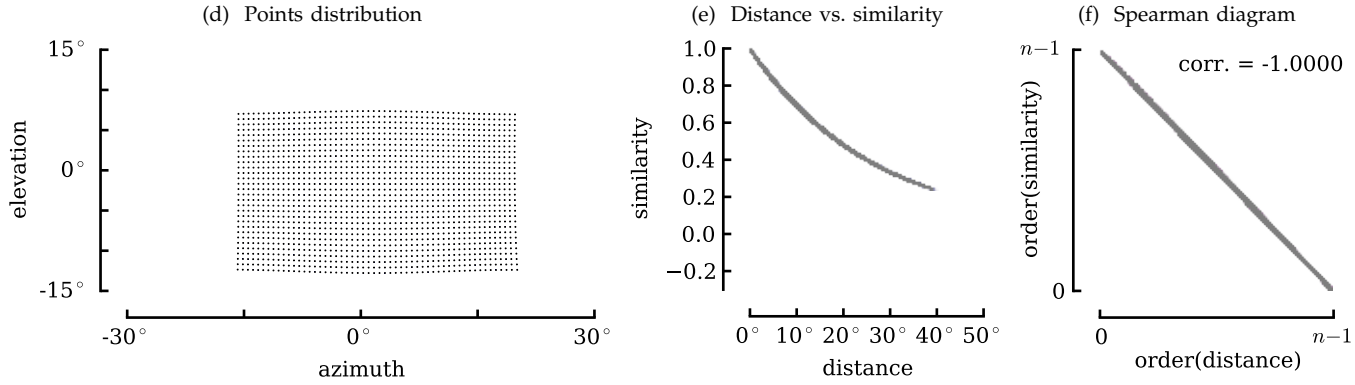


Fig. B-4: FLIP / f_{exp}

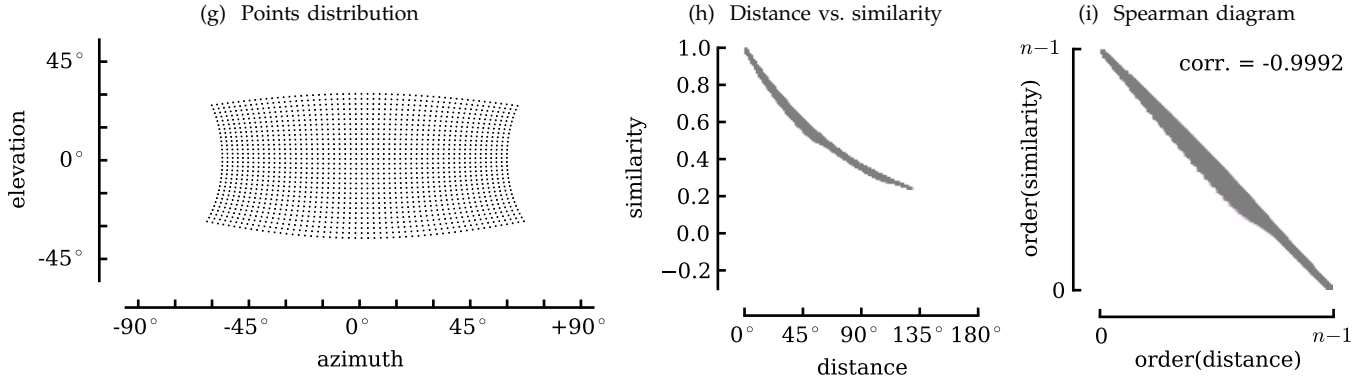
Ground truth



SKv+w



SKv



$\text{MDS}_{\mathbb{S}^n}$

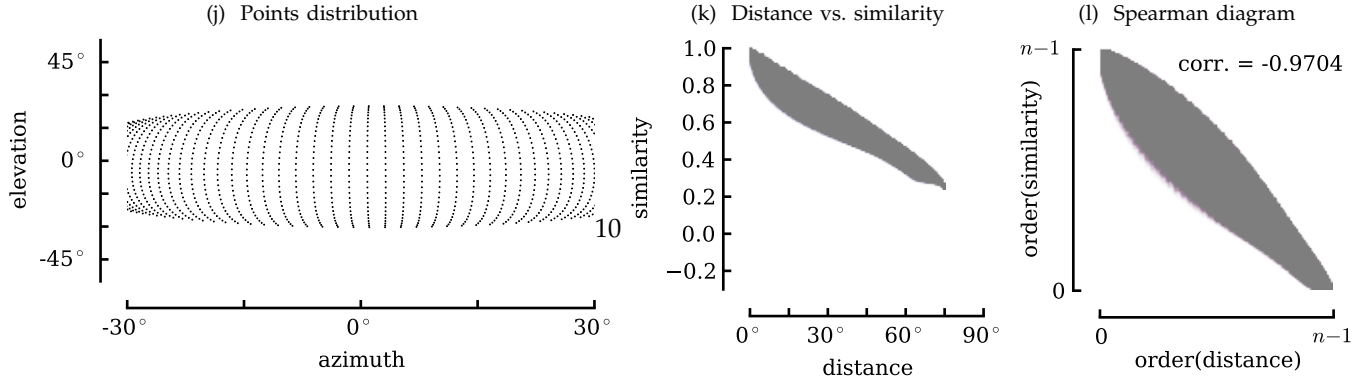
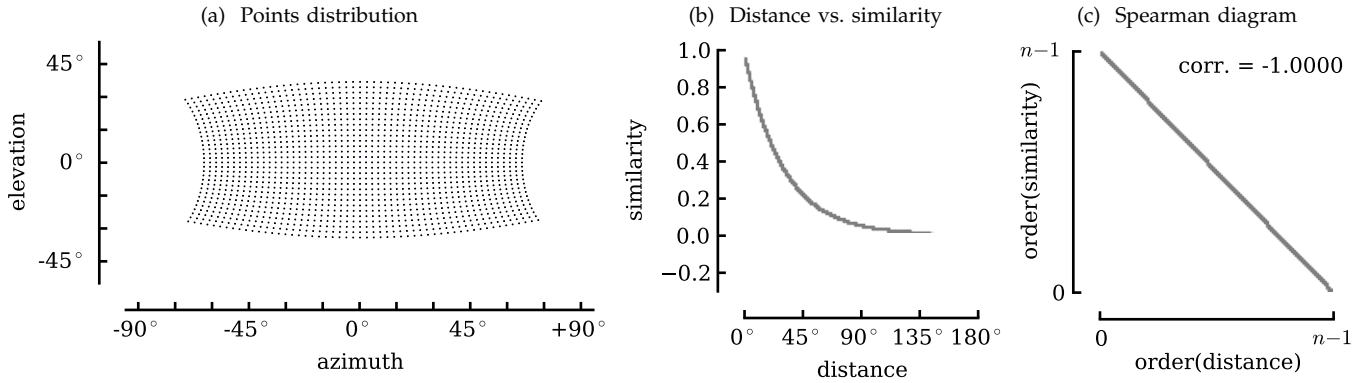
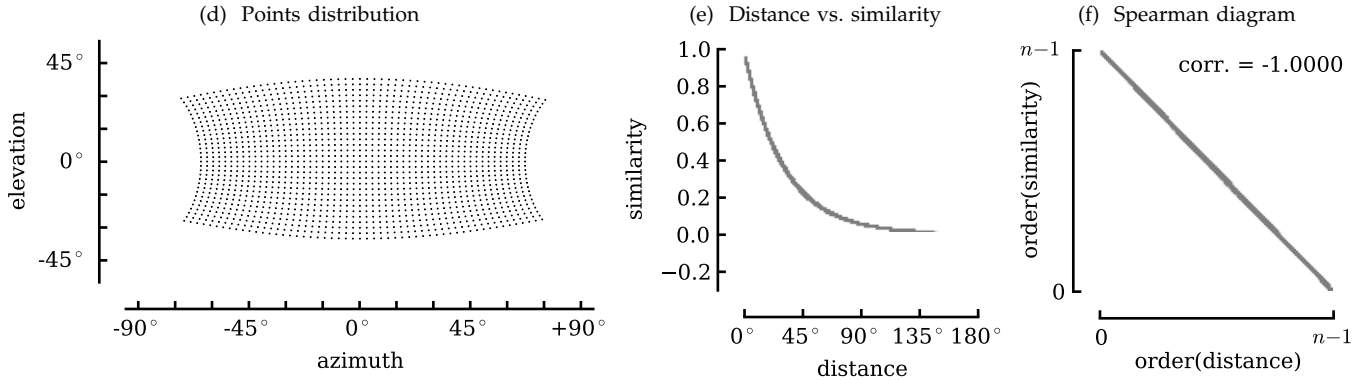


Fig. B-5: GOPRO/f_{exp}

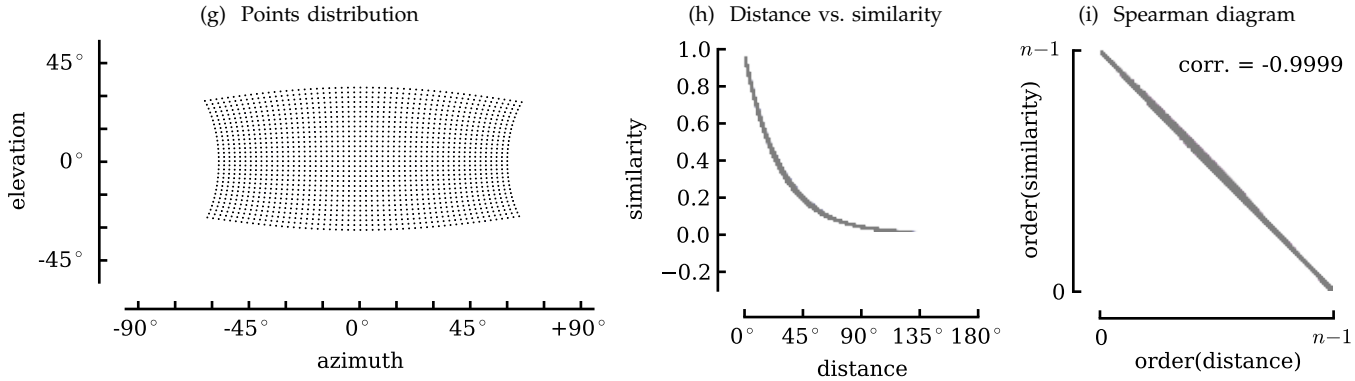
Ground truth



SKv+w



SKv



$\text{MDS}_{\mathbb{S}^n}$

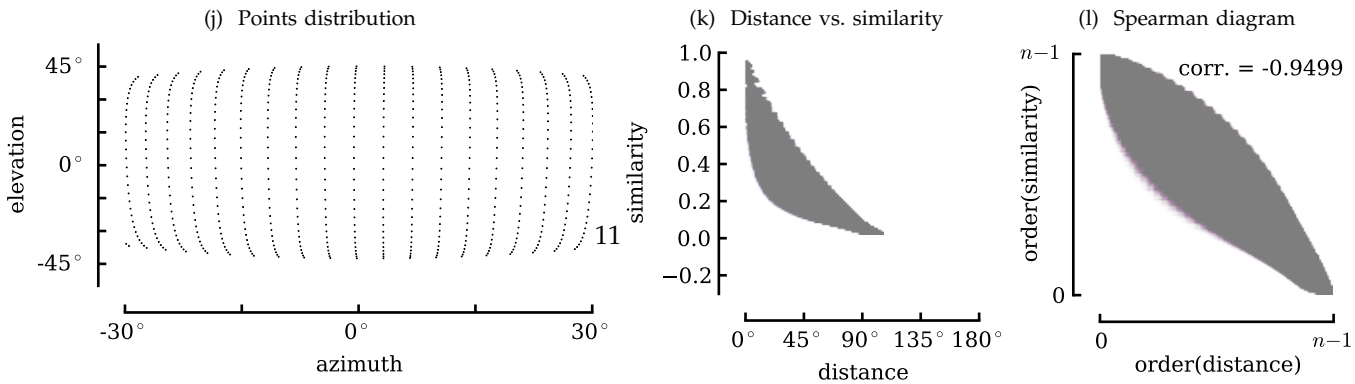
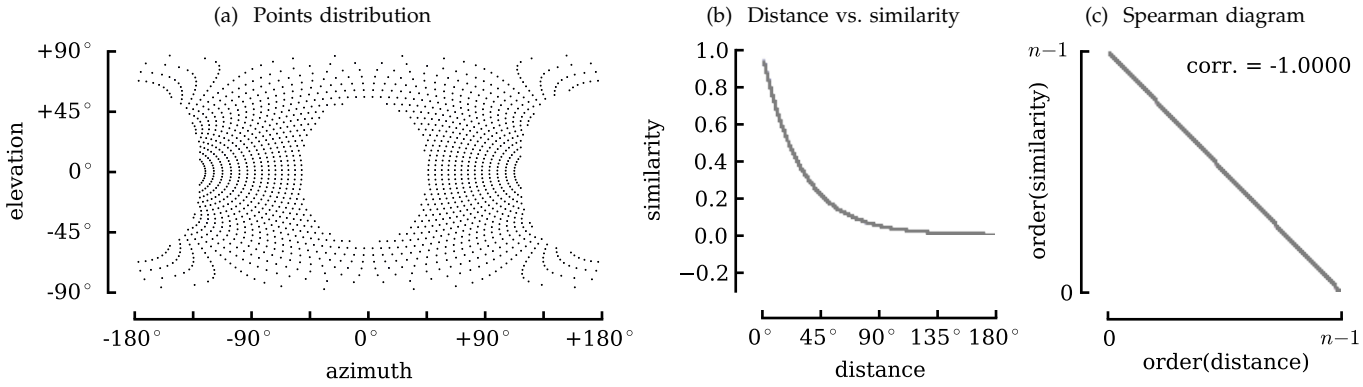
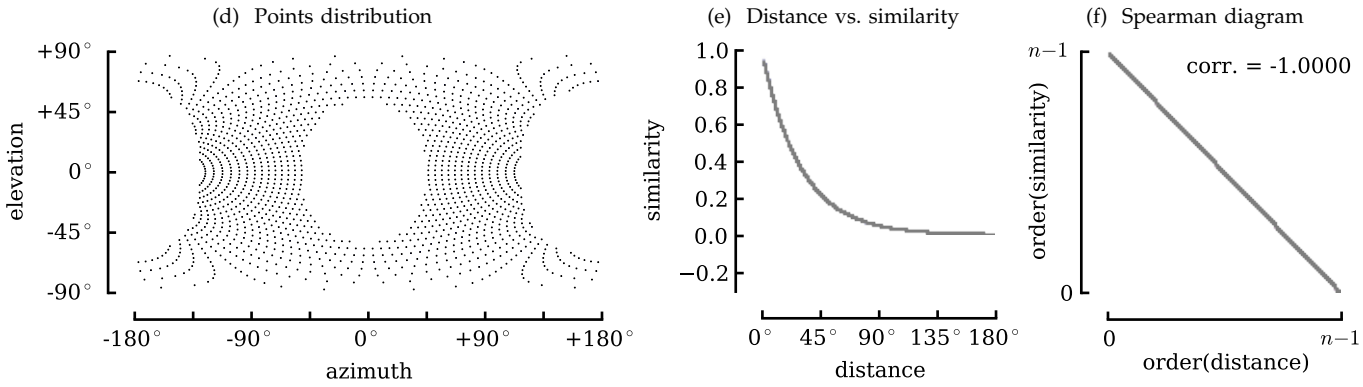


Fig. B-6: OMNI/f_{exp}

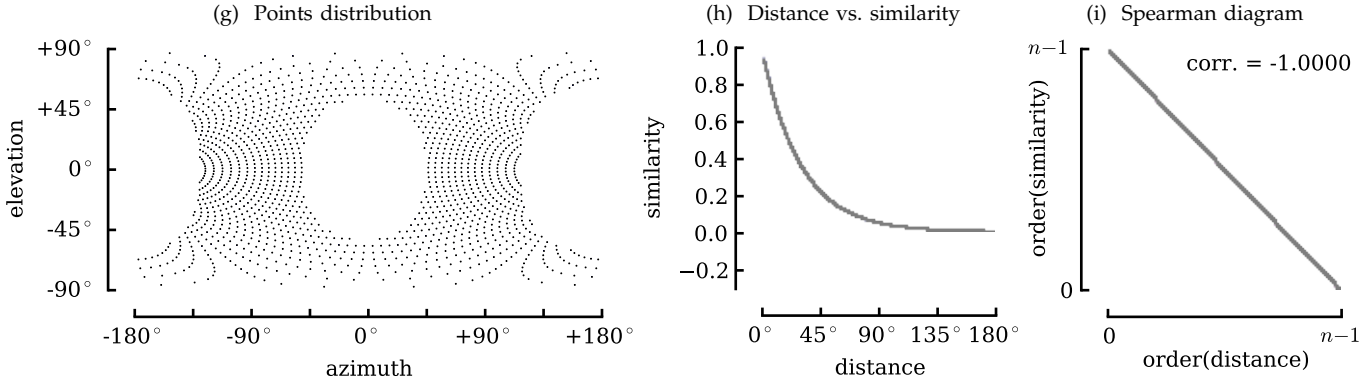
Ground truth



SKv+w



SKv



$\text{MDS}_{\mathbb{S}^n}$

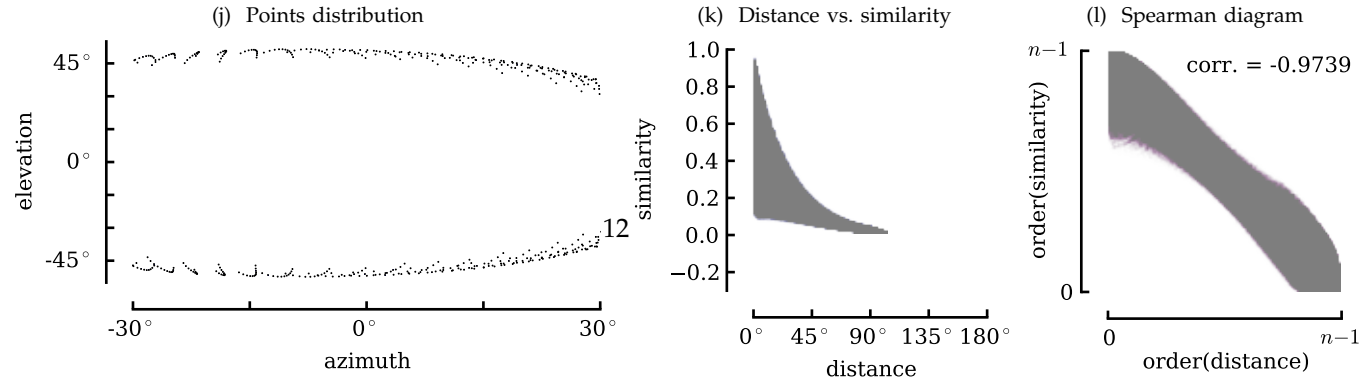
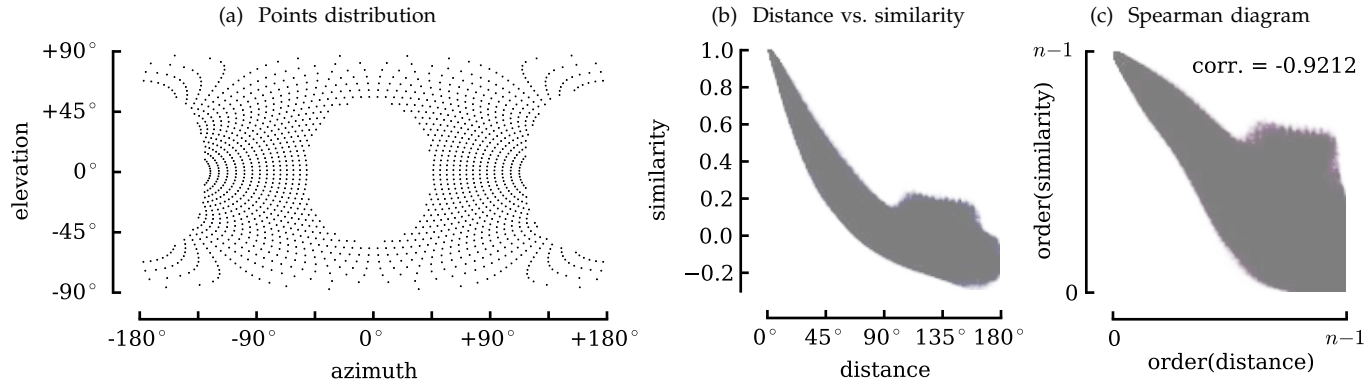


Fig. B-7: OMNI/corr$(c(y))$

Ground truth



SKv+w

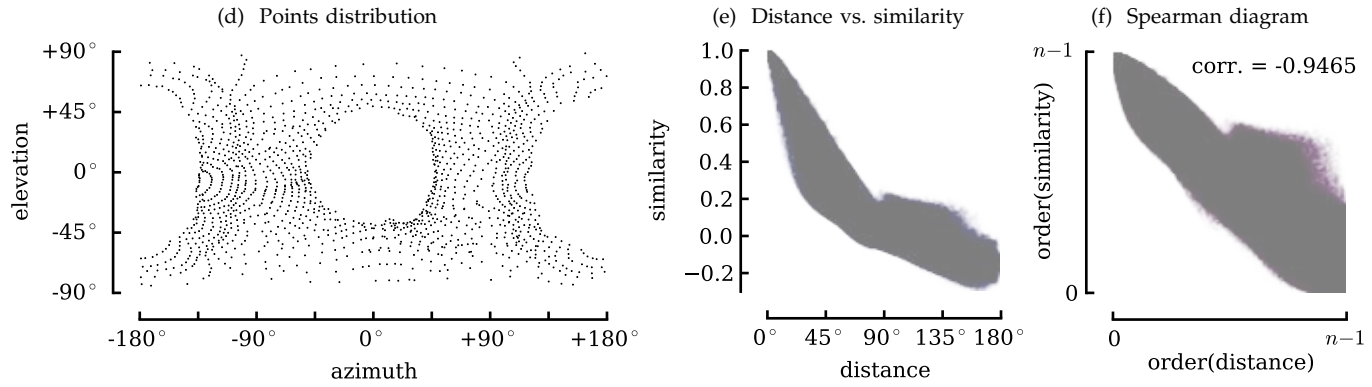
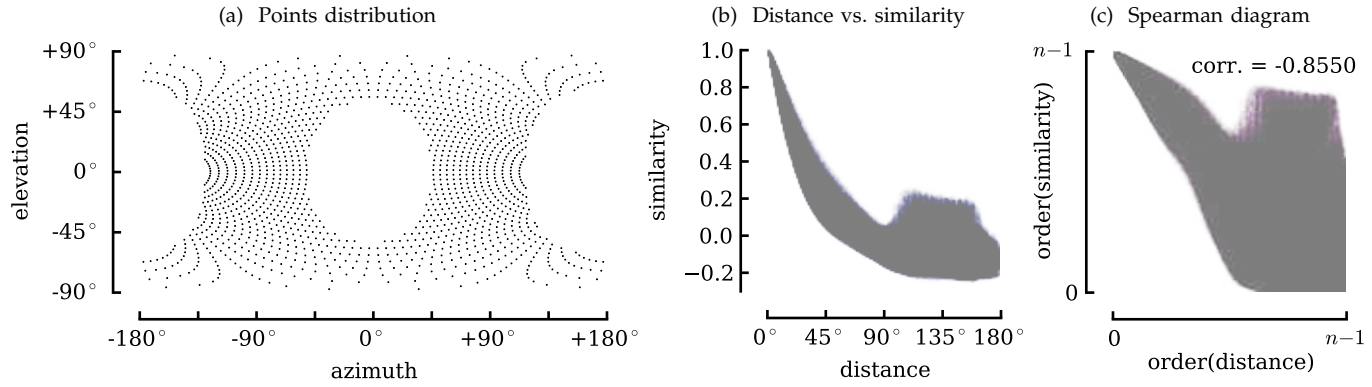


Fig. B-8: OMNI/corr($\dot{y}$))

Ground truth



SKv+w

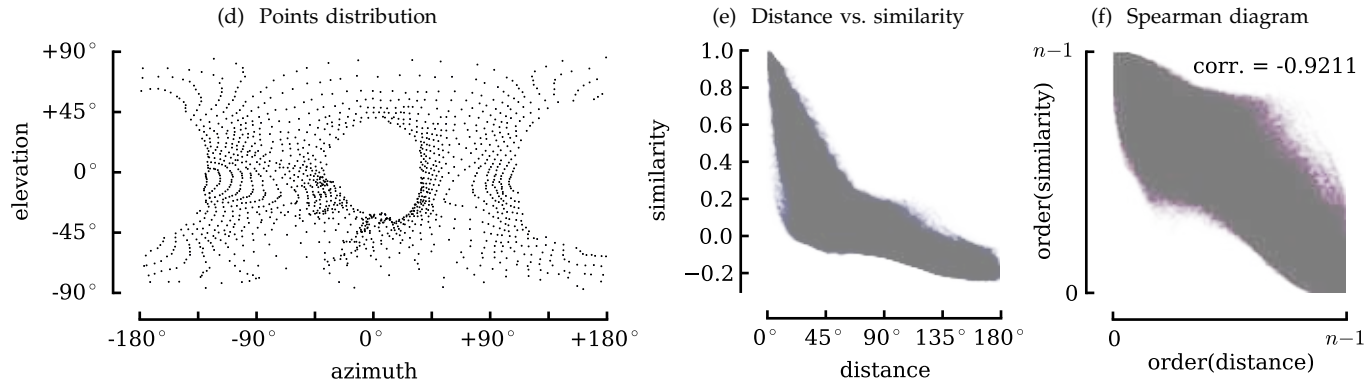
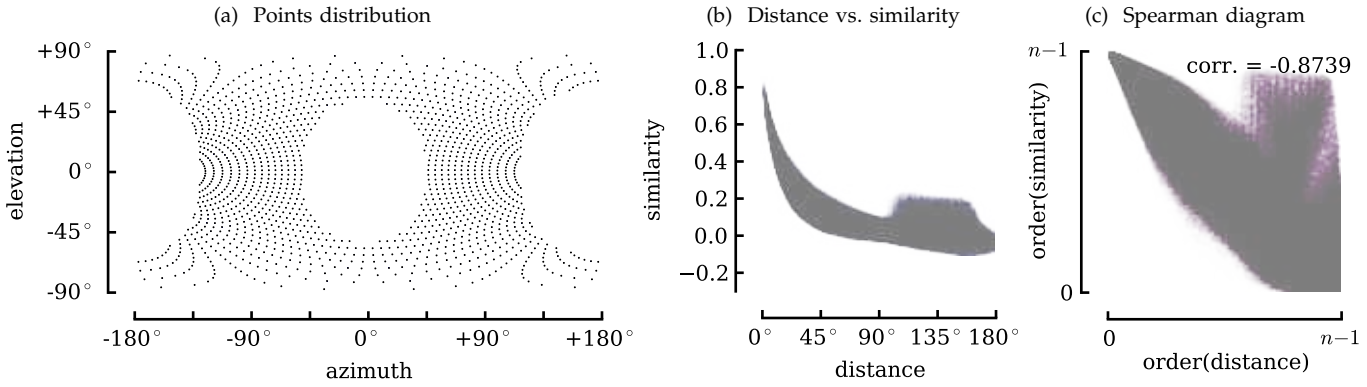


Fig. B-9: OMNI/corr(sgn($\dot{y}$))

Ground truth



SKv+w

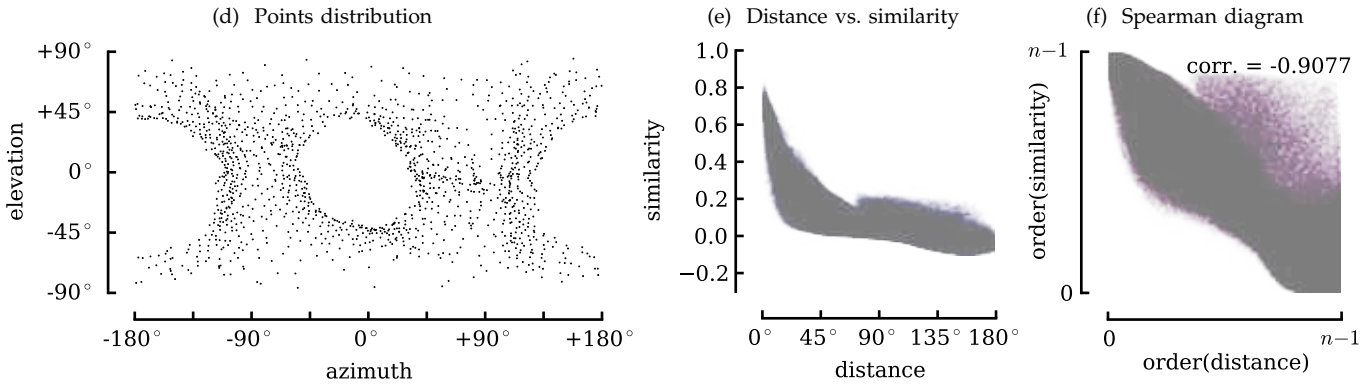
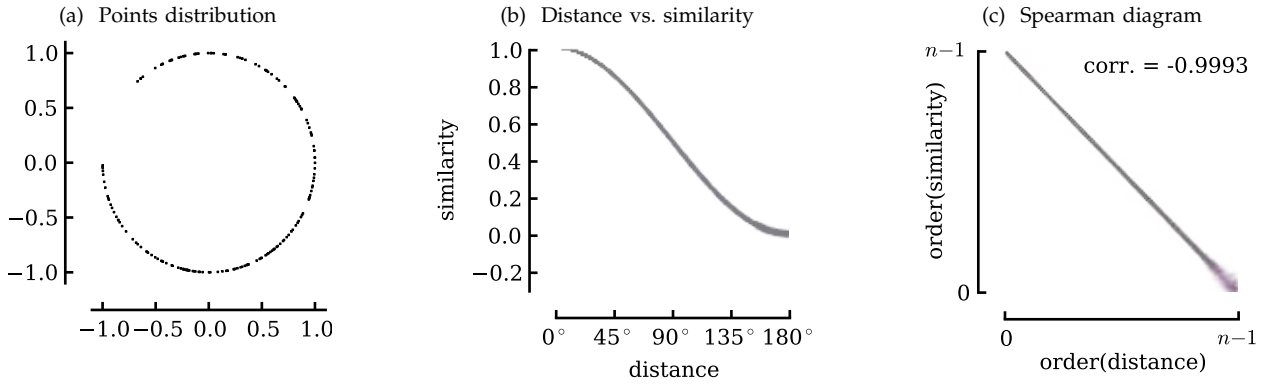
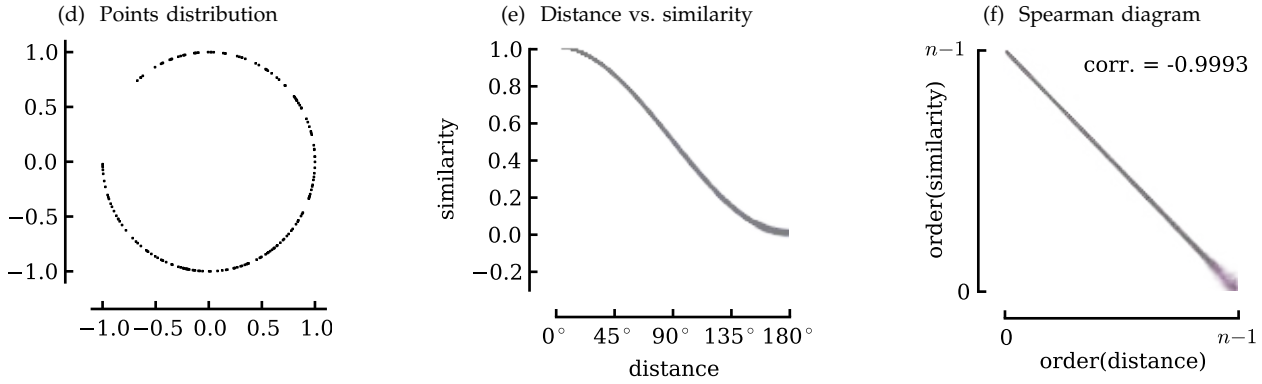


Fig. B-10: Random distribution on $\mathbb{S}^2$, 315deg fov, f_{lin}

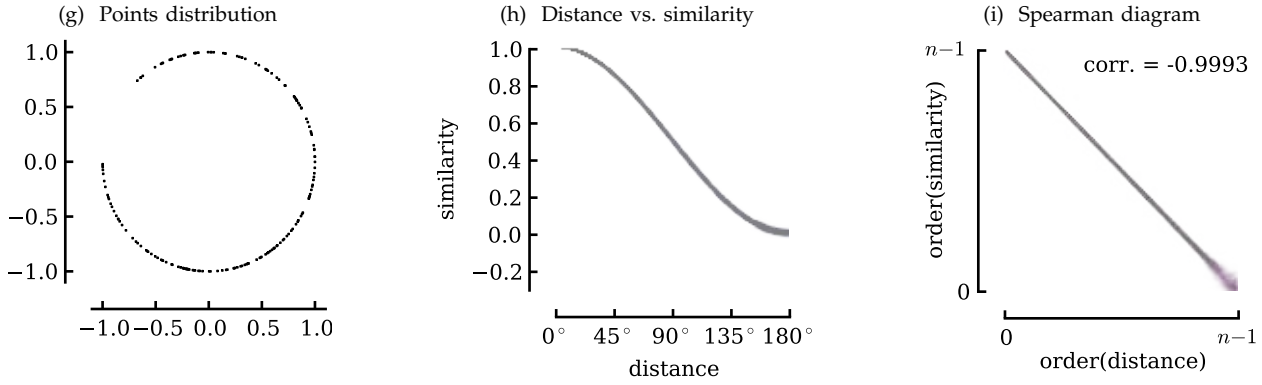
Ground truth



SKv+w



SKv



$\text{MDS}_{\mathbb{S}^n}$

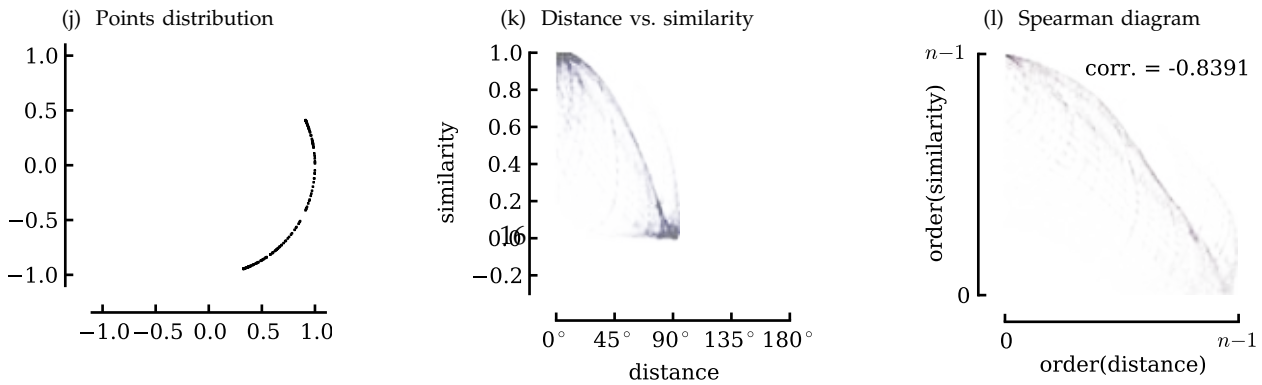
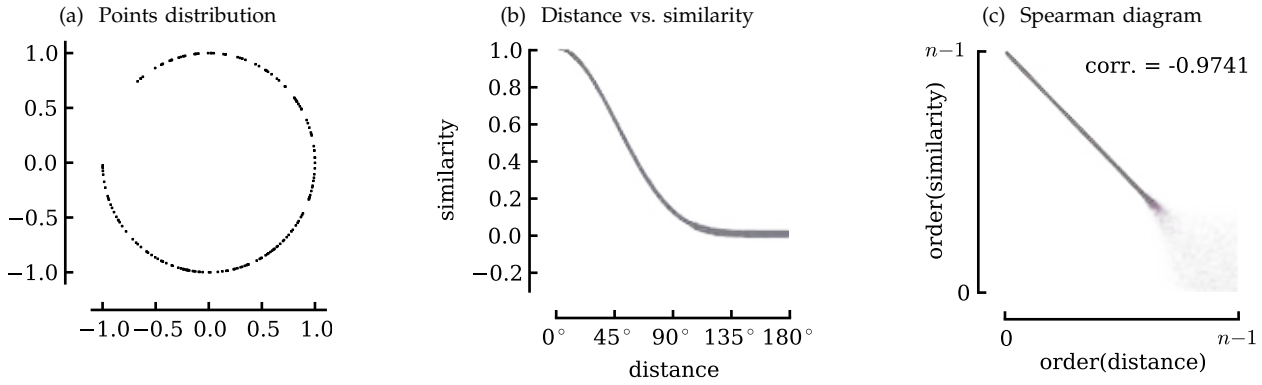
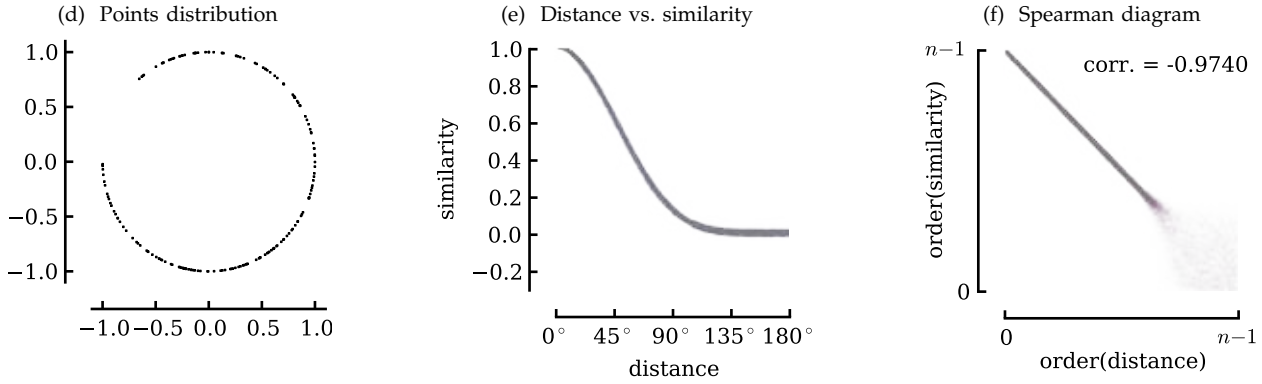


Fig. B-11: Random distribution on $\mathbb{S}^2$, 315deg fov, f_{smooth}

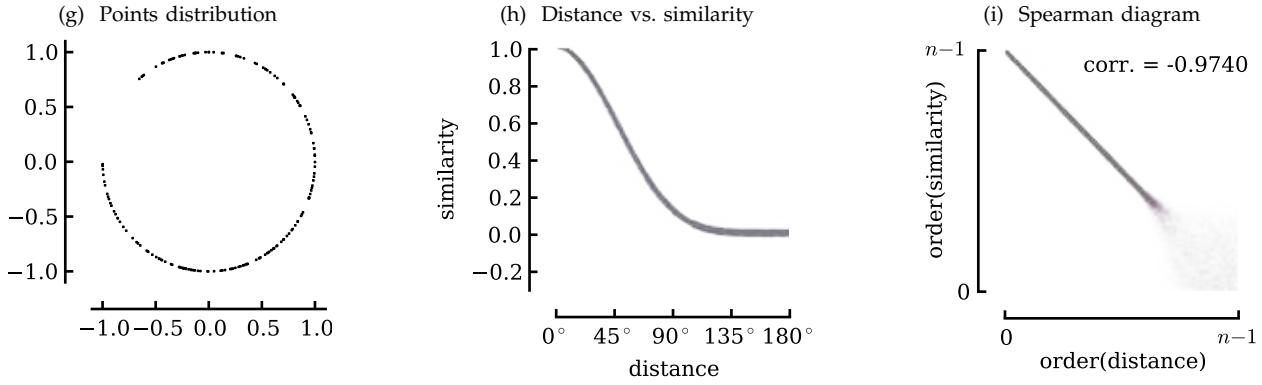
Ground truth



SKv+w



SKv



$\mathsf{MDS}_{\mathbb{S}^n}$

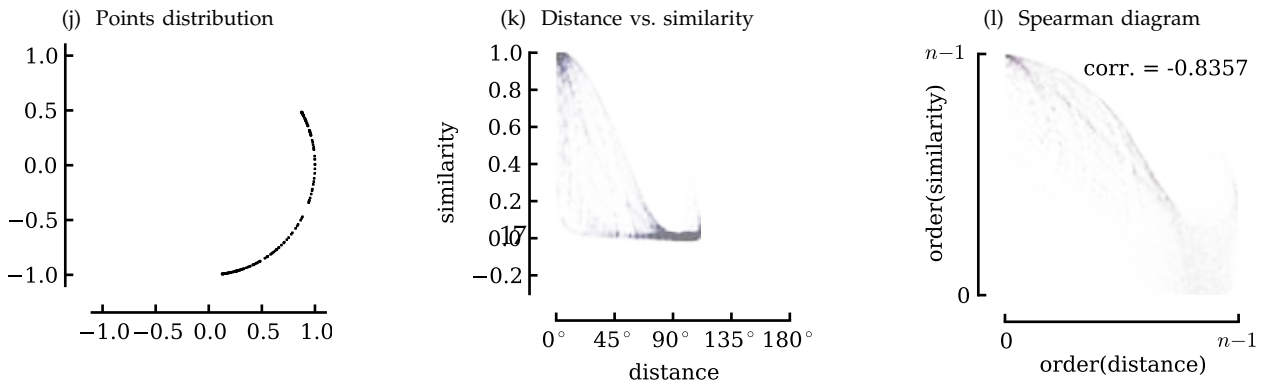
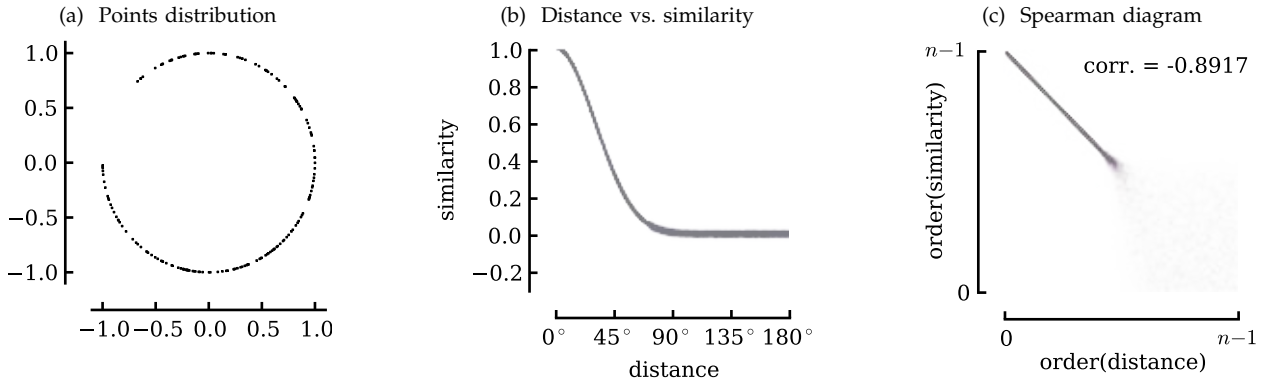
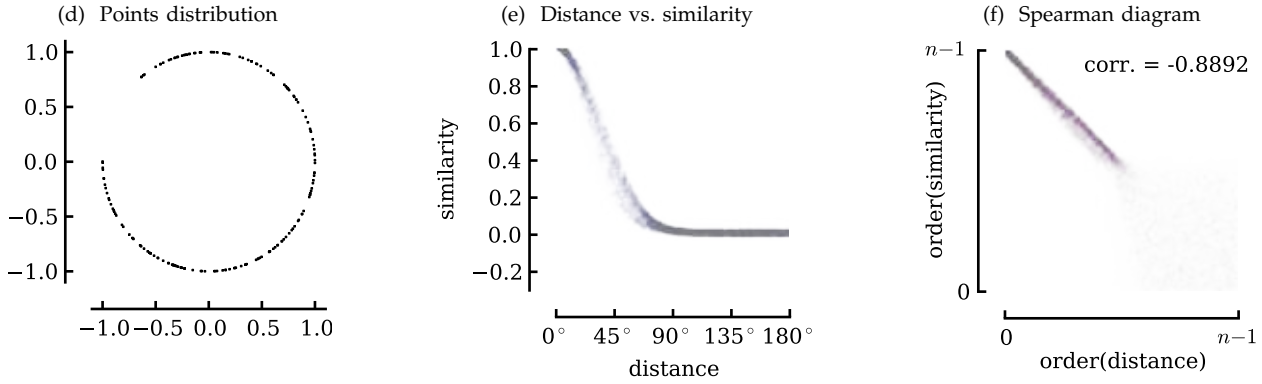


Fig. B-12: Random distribution on $\mathbb{S}^2$, 315deg fov, f_{steep}

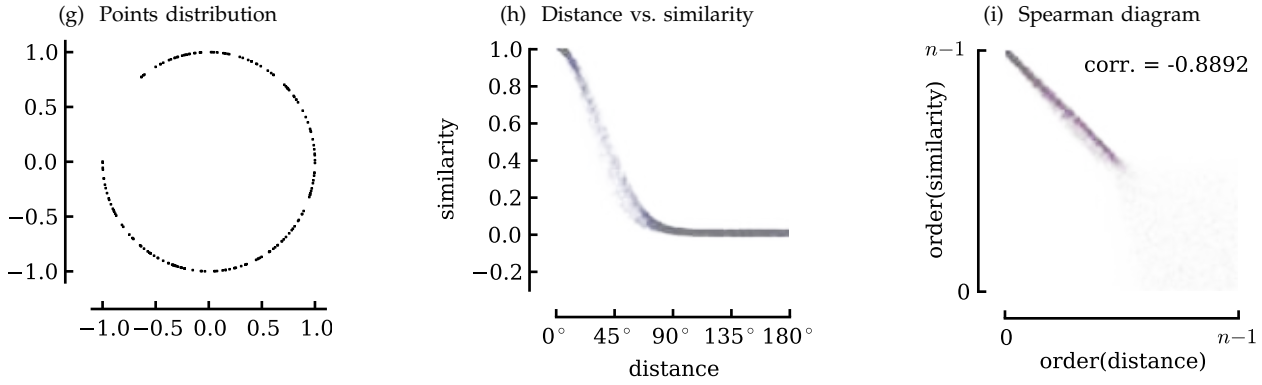
Ground truth



SKv+w



SKv



$\text{MDS}_{\mathbb{S}^n}$

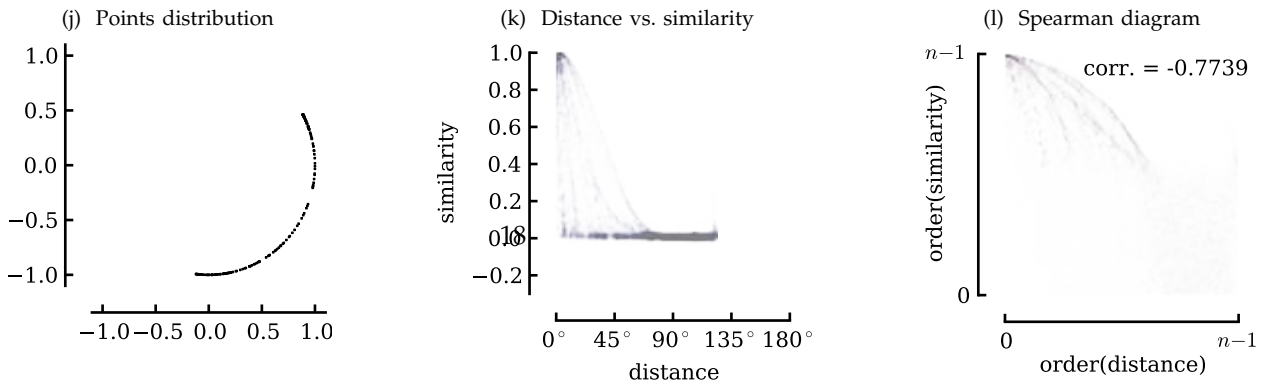
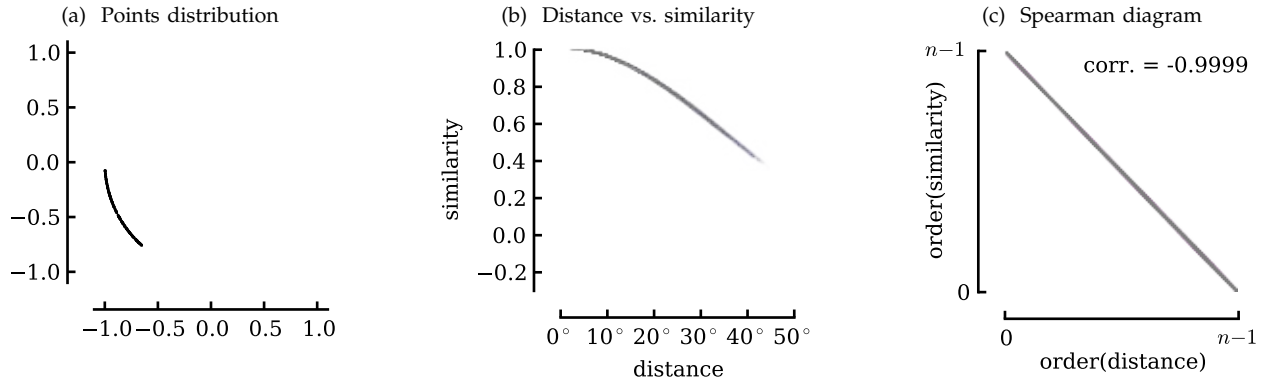
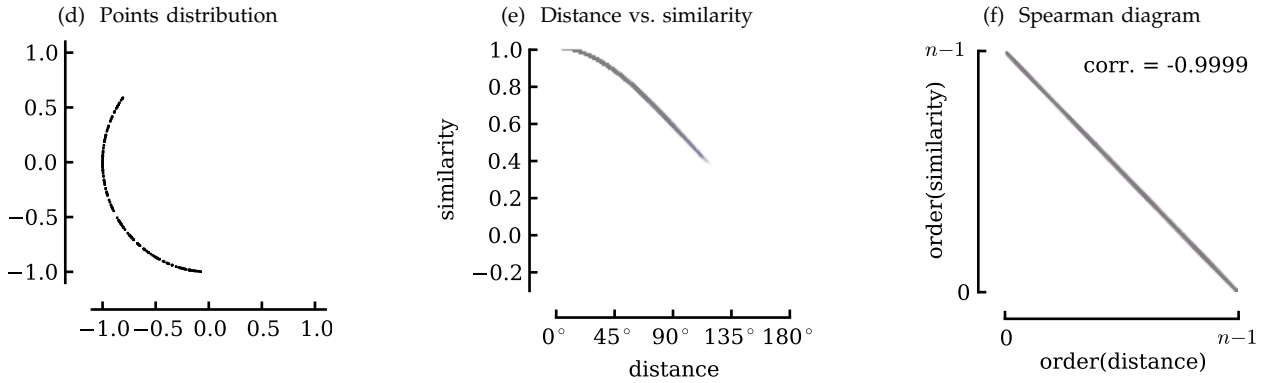


Fig. B-13: Random distribution on $\mathbb{S}^2$, 45deg fov, f_{smooth}

Ground truth



SKv



$\text{MDS}_{\mathbb{S}^n}$

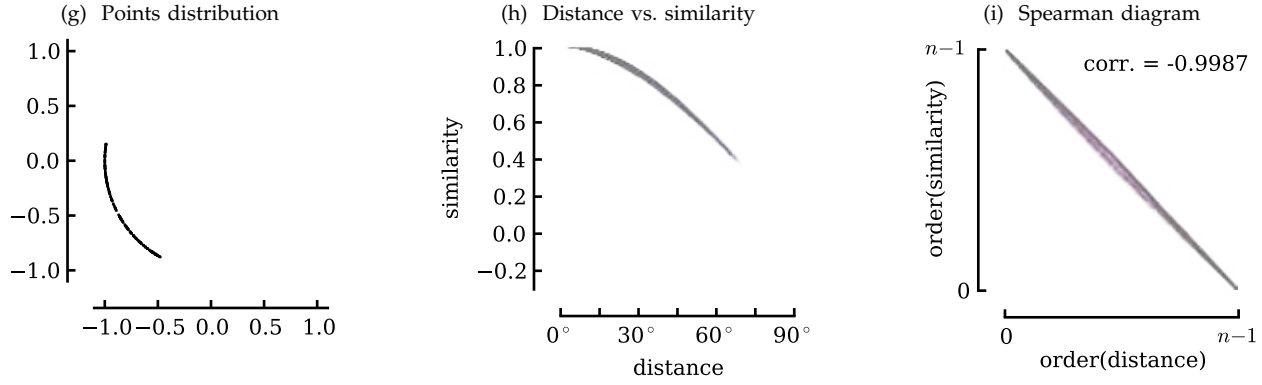
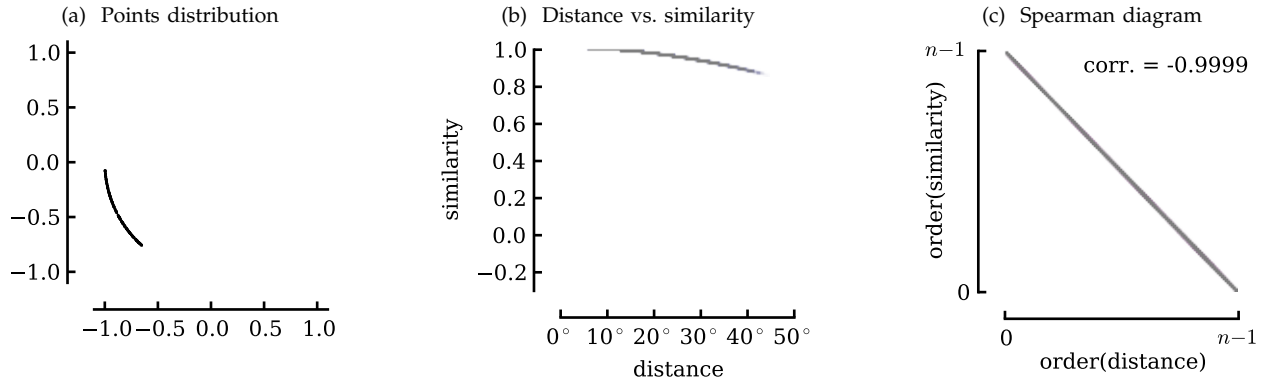
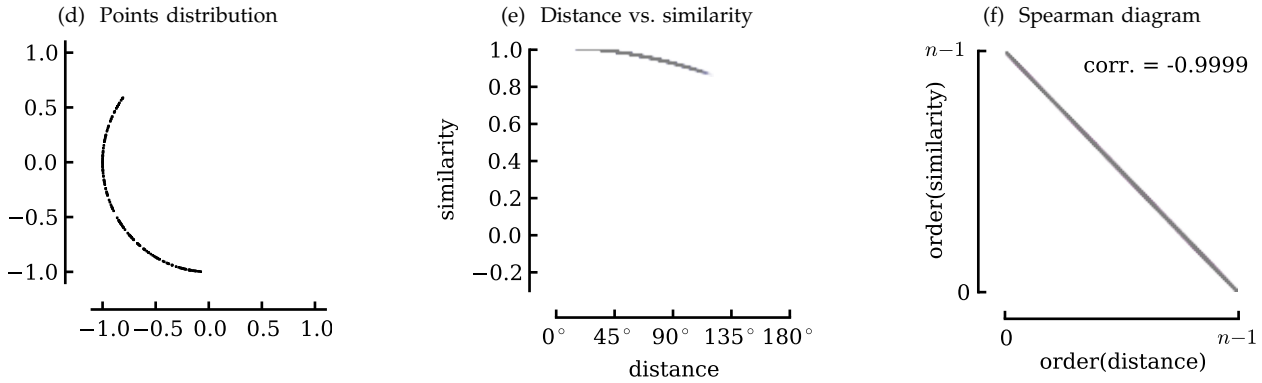


Fig. B-14: Random distribution on $\mathbb{S}^2$, 45deg fov, f_{lin}

Ground truth



SKv



$\mathsf{MDS}_{\mathbb{S}^n}$

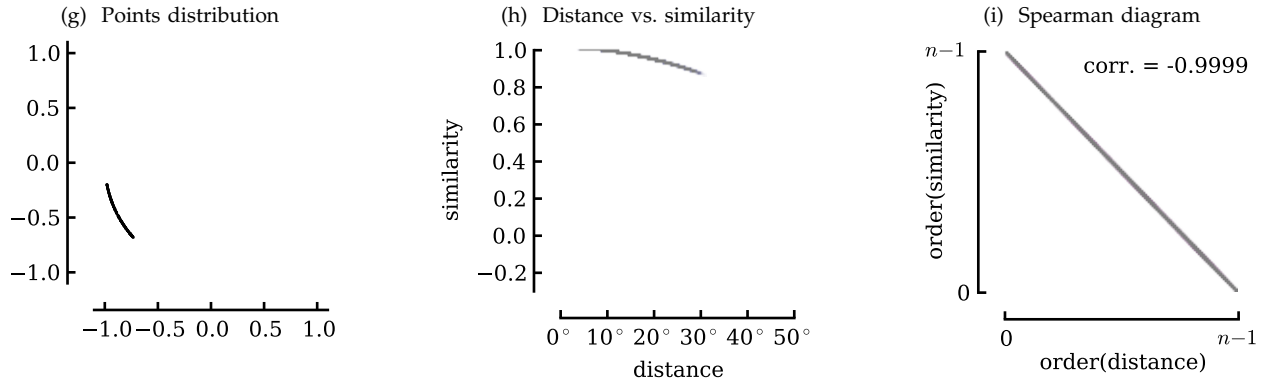
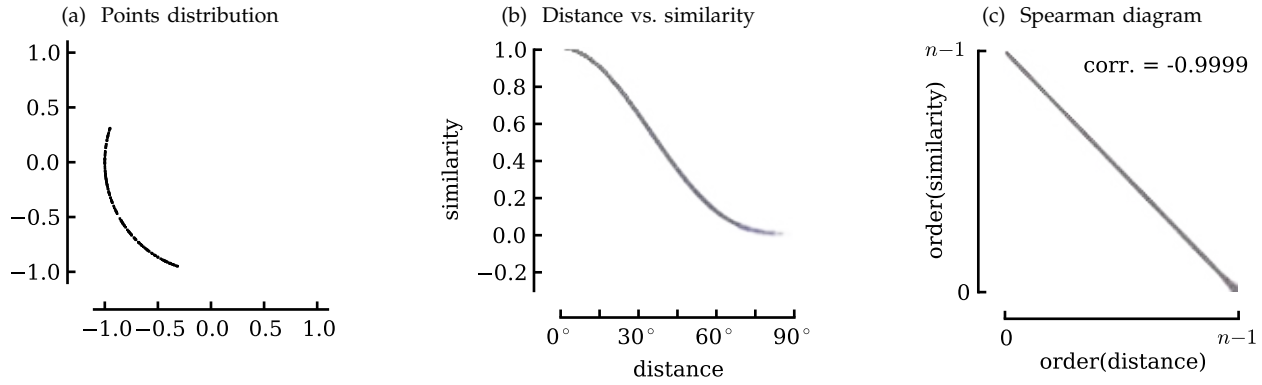
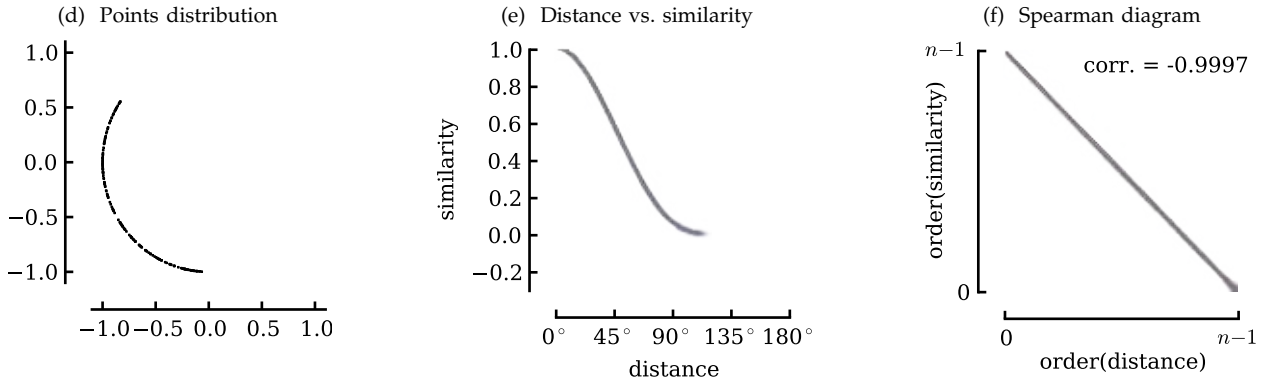


Fig. B-15: Random distribution on $\mathbb{S}^2$, 45deg fov, f_{smooth}

Ground truth



SKv



$\text{MDS}_{\mathbb{S}^n}$

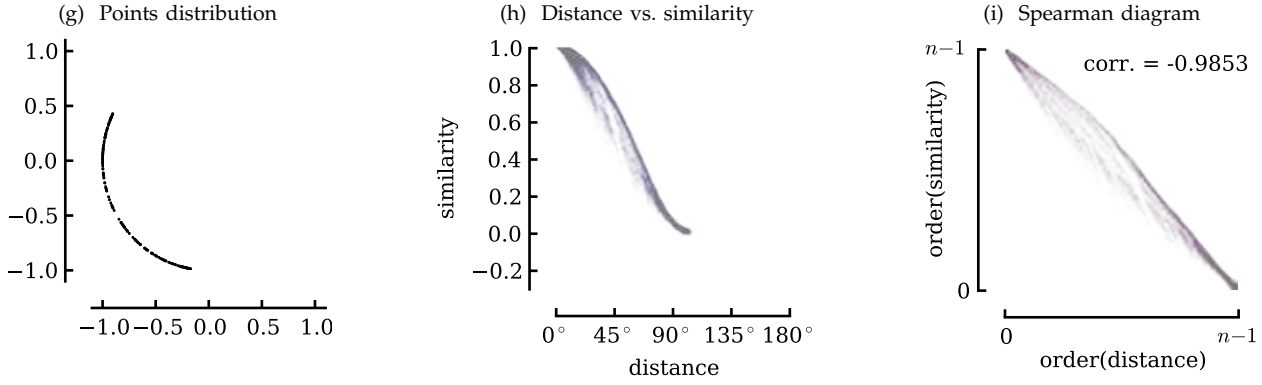
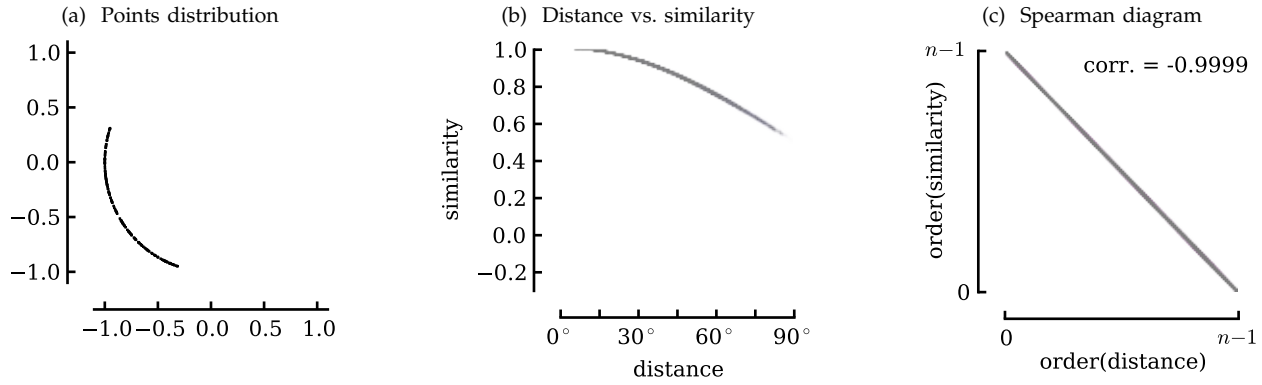
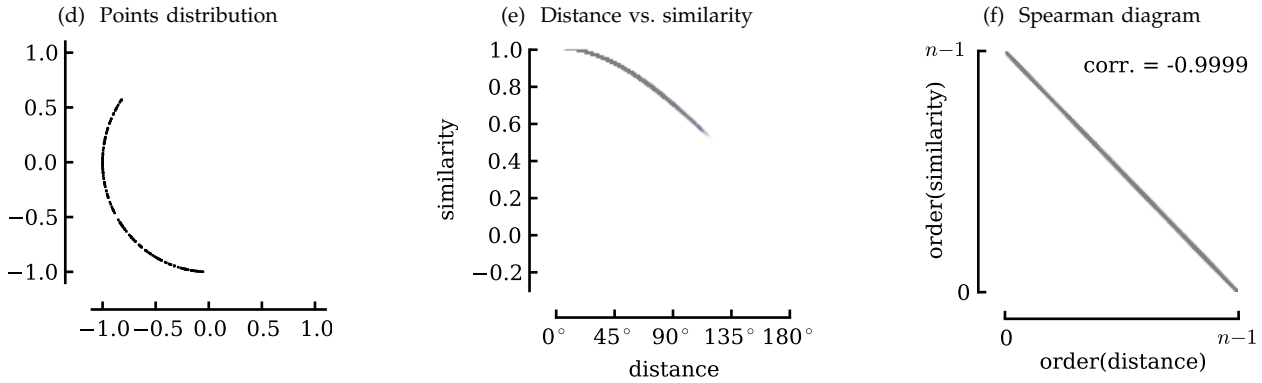


Fig. B-16: Random distribution on $\mathbb{S}^2$, 45deg fov, f_{lin}

Ground truth



SKv



$\text{MDS}_{\mathbb{S}^n}$

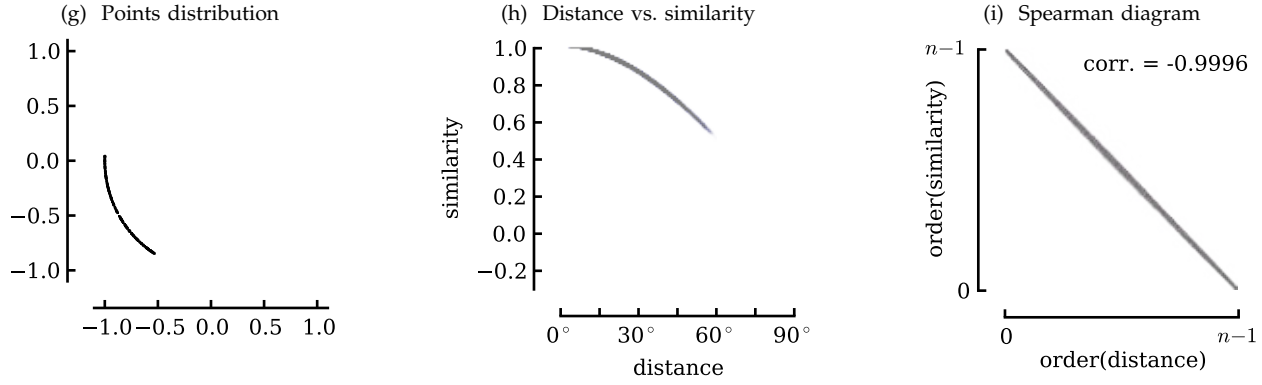
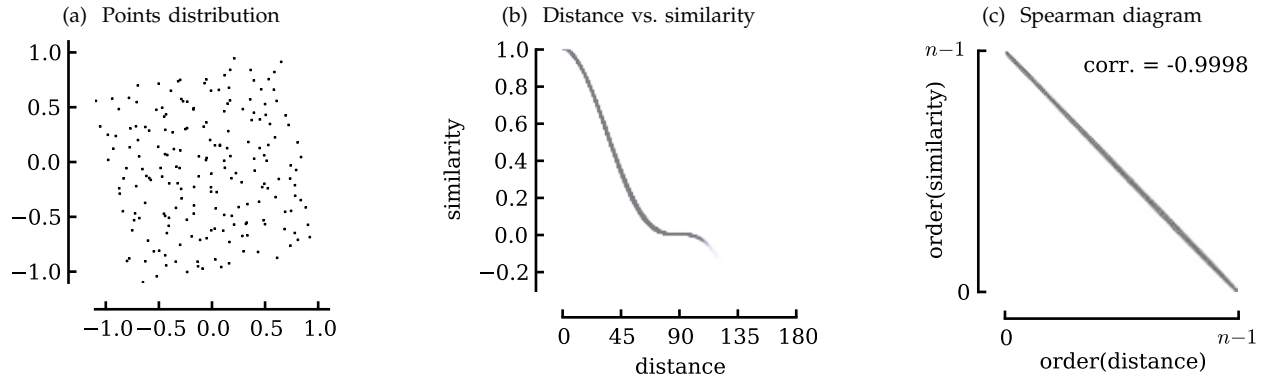
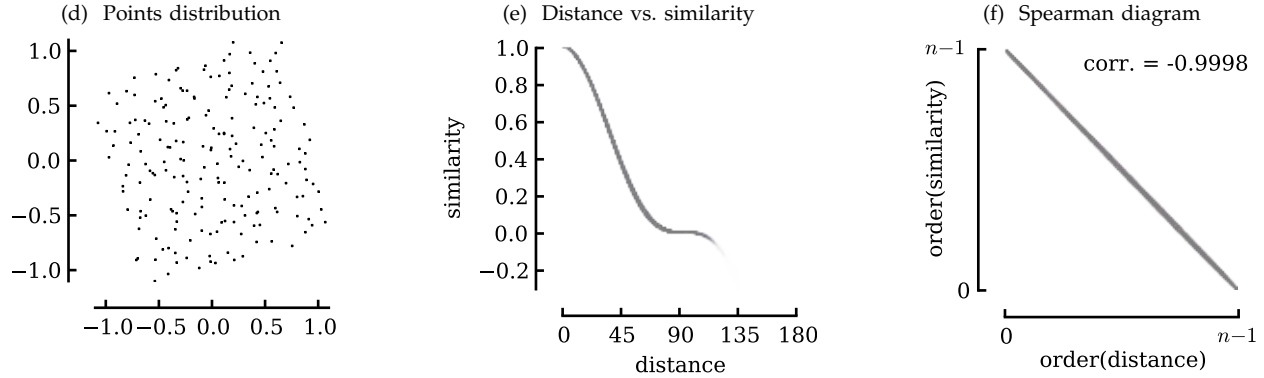


Fig. B-17: Random distribution on a square, f_{smooth}

Ground truth



SKv



$\mathsf{MDS}_{\mathbb{R}^n}$

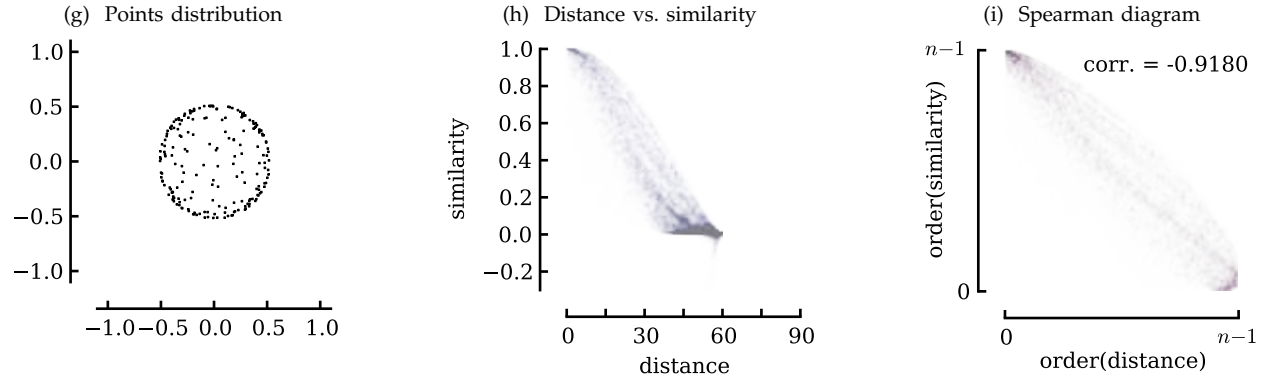
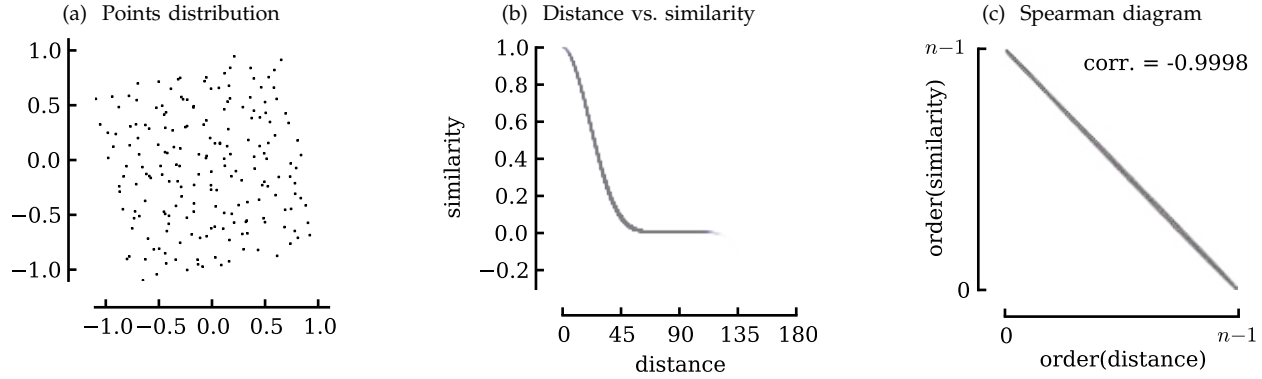
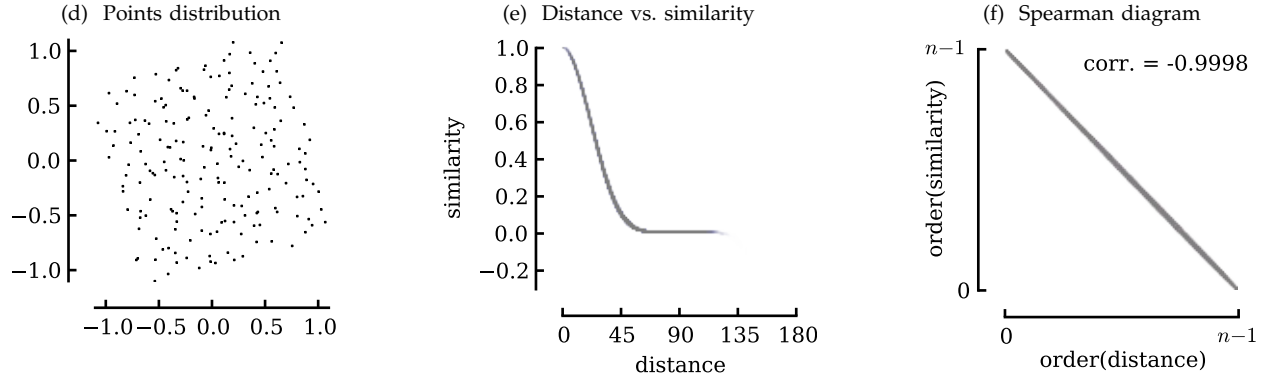


Fig. B-18: Random distribution on a square, f_{steep}

Ground truth



SKv



$\text{MDS}_{\mathbb{R}^n}$

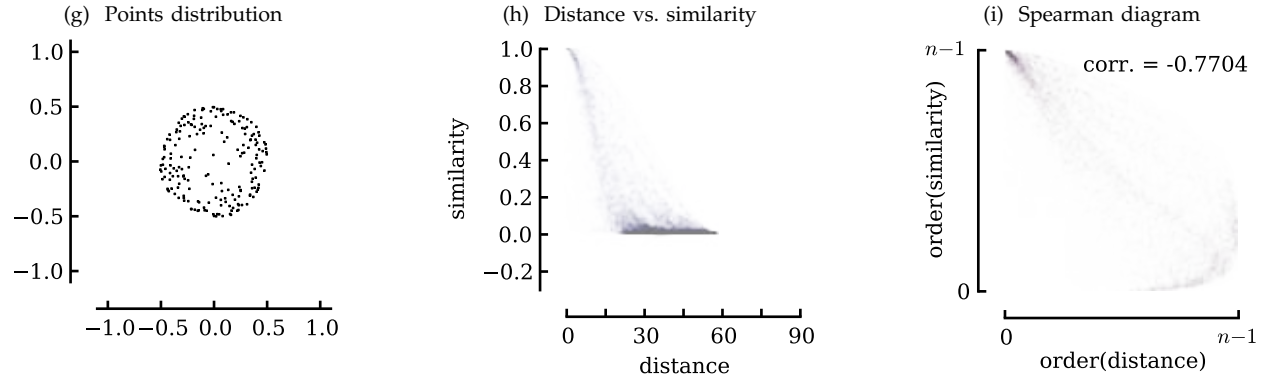
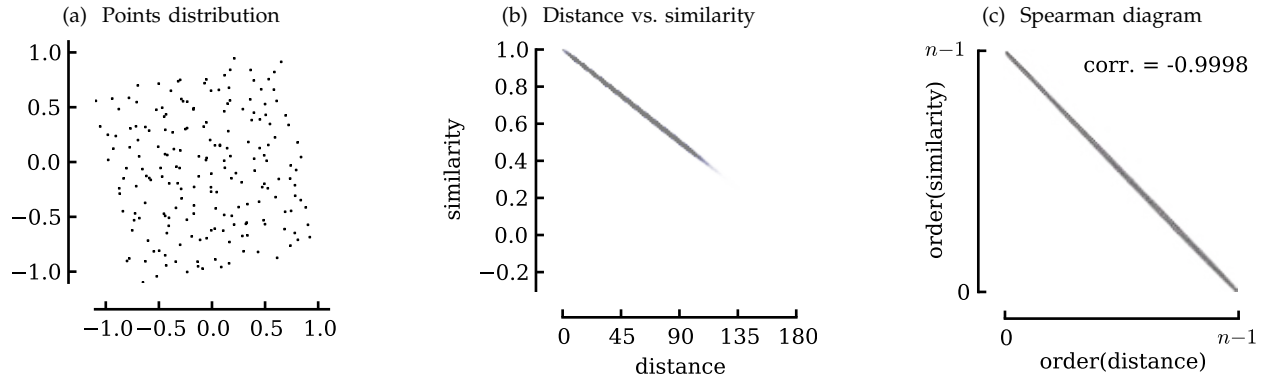
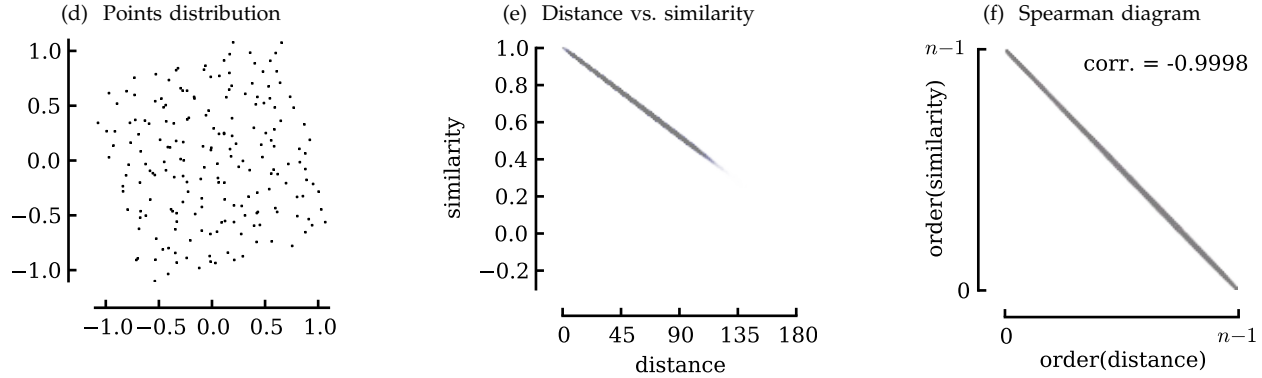


Fig. B-19: Random distribution on a square, f_{lin}

Ground truth



SKv



$\text{MDS}_{\mathbb{R}^n}$